IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 12, 2019, accepted February 22, 2019, date of publication February 28, 2019, date of current version March 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902166

Efficient Topology Reconstruction via Machine Learning Based Traffic Patterns Recognition in Optically Interconnected Computing System

CEN WANG[1], (Student Member, IEEE), HONGXIANG GUO[1], XIONG GAO[1], YANHU CHEN[1,2], YINAN TANG[1,3], AND JIAN WU[1], (Member, IEEE)

[1]State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Quantum Computing Group, IBM Research, Beijing 100193, China
[3]College of Engineering and Computer Science, Syracuse University, Syracuse, NY 13235, USA

Corresponding author: Hongxiang Guo (hxguo@bupt.edu.cn)

This work was supported in part by the Natural National Science Foundation of China (NSFC) under Grant 61331008 and Grant 61471054.

ABSTRACT The traffic flows in parallel computing systems show clustered, correlative nature and the flows are always latency-sensitive. These flows have been abstracted as ''Coflow'' to pursue overall optimization. Concurrent Coflows on the network show very novel traffic patterns. On the other hand, multiple optical interconnection network architectures have been proposed to enable the traffic adaption topology reconstructions. Nevertheless, topology reconstruction strategies are application-agnostic, and their optimization objective of network performance cannot meet the Coflow demand. In order to exert the flexibility of optical topology to promote the performance of parallel computing application by Coflow acceleration, the traffic patterns are preferred to be well recognized and then an adaptive topology is generated accordingly. To avoid further complex, such recognition is expected to finish without prior knowledge from the application layer. Then, the topology should be reconstructed to minimize the Coflow completion time. To implement these procedures, we proposed a traffic pattern-aware topology reconstruction strategy. Our strategy first combines CNN and spectral clustering to realize the traffic patterns awareness. And then, the genetic searching algorithm is used to mind the proper topology. Based on real traffic trace from Facebook computing application, large-scale simulations have verified the efficiency of such a strategy by lowering the completion time of computing jobs. In addition, the experimental demonstration has confirmed the conclusions.

INDEX TERMS Traffic pattern, topology, optical interconnections, computing system, Coflow.

I. INTRODUCTION

Hadoop [1] and Spark [2] are the most popular distributed parallel computing systems based on data centers (DCs). In these computing systems, parallel data should be transmitted across the racks of data center networks (DCNs). The traffic pattern in each job is a group of clustered ''all-to-all'' connections. And the traffic flows in a job are defined as a ''Coflow'' in [3]. The flows in a certain Coflow are latency-sensitive and are expected to finish within the same barriers, as depicted in FIGURE 1(a). In a summary, the traffic pattern in a job shows clustered and correlative nature. Usually, several computing jobs are concurrent on the network [4]. Thus, on the rack-level, the traffic patterns across the whole network can be described as FIGURE 1 (b).

On the other side, the optical interconnection networks have been introduced into DCs in recent years. Despites the ultra-large bandwidth and low switching latency, the topology flexibility is also deployed to support different traffic patterns. In the optical DCNs such as Helios [5], C-through [6] and OSA [7], the flexibility of network is explored by dynamically reconfiguring light paths via algorithm like *b-matching* [8] to maximize network throughput. However, such reconstruction strategy is application-agnostic, and sometimes may deteriorate the application performance because of the ignorance of clustered and correlative nature within traffic patterns. On improper topology, the traffic with clustered and correlative connections are separated by multiple hops or are assigned with limited

The associate editor coordinating the review of this manuscript and approving it for publication was Lin Wang.

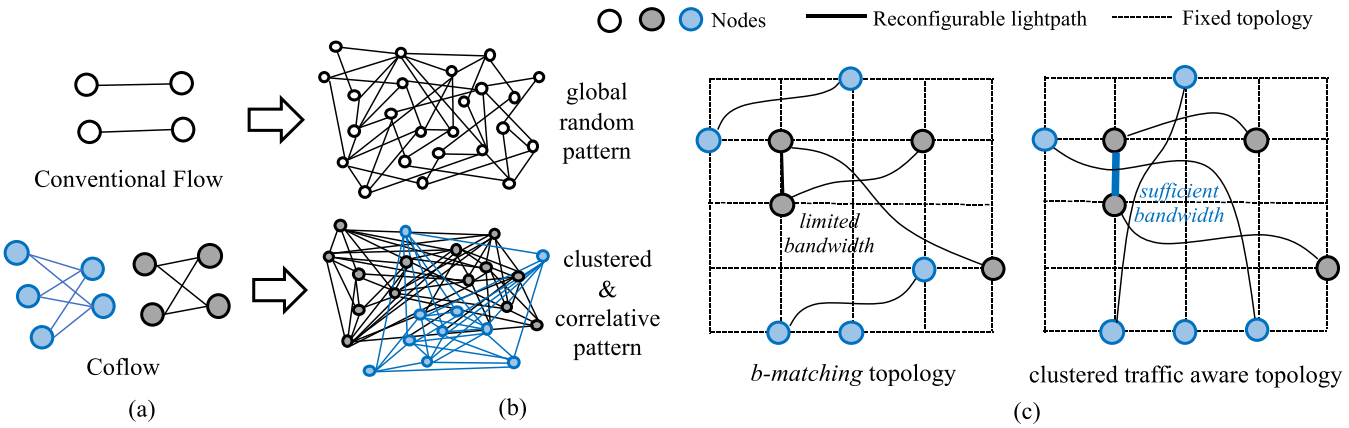

FIGURE 1. Coflow, traffic patterns and the improper topology.

bandwidth, as shown in FIGURE 1 (c). In order to exert the flexibility of optical DCNs to optimize application performance, the topologies are advocated to be reconstructed to support the diverse clustered traffic patterns. And then minimizing the Coflow completion time (CCT) becomes more essential objective than maximizing total throughput in reconstruction strategies.

There are previous works that have demonstrated the reconfigured topology to support the computing jobs, such as the ''a table for two topologies'' [9] and the ''OvS'' [10]. They both can accelerate the job completion time by adjusting their topologies. But these works have only verified their capability in traffic adaption. To drive the ''capability'' to become ''efficiency'', there are still problems left to be solved that how to recognize the traffic patterns in computing system, and what is the proper topology strategy.

In this paper, we propose such a universal clustered traffic -aware topology reconstruction strategy to contribute the topology efficiency. It can be deployed on any optical network which has ability to reconstruct topology to benefit the traffic patterns. To implement such reconstruction strategy, the clustered traffic should be firstly recognized. Clustered traffic recognition is actually to identify the number of computing jobs, and which racks a job is running on. The traffic pattern can be known by job information acquired from the application, but this may add more extra development costs. Thus, the traffic pattern is preferred to be recognized without prior knowledge. Nevertheless, it is such a challenge to do the accurate recognition in this way for the correlative yet various traffic patterns. So as to solve this problem, the machine learning method, i.e. the convolutional neural network (CNN) combined with the spectral clustering are utilized. And then based on the recognized traffic patterns, the proper topology is calculated via the genetic algorithm to minimize the CCTs.

The traffic patterns -aware topology strategy is evaluated by both the simulation and experiment way. The simulation is based on large-scale networks. These large-scale networks can be regarded as fixed topology combined with reconfigurable light paths. The different fixed topology span on a Tree network, a basic Lattice network [11], a basic Cubic network [12], and a Small-world network [13]. These networks can be logically achieved by optical switching, the approaches are described in Section 2. In the simulation, we use the real Coflow trace from Facebook [14]. Such real data trace is injected in the above four networks. Then, the accuracy of recognition and the CCT promotion are evaluated. And based on the evaluations, the efficiency of such strategy has been verified. In the experiment, the performance of proposed topology strategy is demonstrated on the network with fixed Tree topology and the with fixed Small-world topology. The results have confirmed its efficiency by lowering the job completion times (JCTs).

In the rest of the paper, we firstly introduce background and motivations to propose the clustered traffic patterns -aware topology reconstruction in Section 2. Then, the models and algorithms are detailed in Section 3. In Section 4, we illustrate the simulation and analyze the corresponding evaluation results. And finally, the experimental demonstration is described in Section 5.

II. BACKGROUND AND MOTIVATIONS

A. TRAFFIC PATTERNS IN DISTRIBUTED COMPUTING SYSTEM

The traffic pattern of a computing job has clustered and correlative nature. In the large-scale computing system, for instance, based on DC, many jobs have to treat ultra-large data processing. Due to the computing and storage limitation, usually these jobs are assigned resources across the racks, so that the inter-rack communications are required. In a job, the traffic among racks show ''all-to-all'' pattern. So, the traffic can be regarded as clustered.

Additionally, in a job, the traffic flows between these racks usually start at same time and they are expected to complete with same barriers. Because, only if all the traffic flows finish, the next step of a job will be executed. This is similar to the buckets effect, the traffic flow with longest completion time will determine the CCT. Thus, the traffic pattern in a job also can be concluded as correlative.

According to the job trace from Facebook [14], several jobs are concurrent on the network in a period of time. A job is executed among a group of racks. The traffic inside the group is strong. But there is weak or no inter-group traffic. Different groups may have overlaps. Thus, in a holistic view, the traffic pattern from all the network can be seen as several groups with inside clustered yet correlative connections.

To accelerate the CCT is very essential to optimize the computing system. The CCT can occupy 50% of the JCT. And this fraction will be enlarged by the job scale (i.e. the number of racks that a job is running on) increasing [15].

B. TRAFFIC RECOGNITION APPROACHES

As for the traffic recognition in computing system, e.g. the Hadoop/Spark on DCs. Tightly follow the concept of maximizing the throughputs, the Flyways [16] recognized the large flow and the small flow by counting the packet numbers of a flow. And then the flyways (i.e. the optical paths) are connected for the large flows. In Karuna [17], the flows with or without deadline are recognized according to the option segment in TCP header. And then the resource is scheduled to optimize the mix-flows. In [18], the virtual topology design (VTD) by means of cognition is proposed, this cognition is actually the traffic prediction but rather than patterns recognition. And in CODA [19], the Coflow is recognized via flow-level features. The recognition is to cluster the host-to-host flows into different Coflows. But this method could not provide quick enough recognition of traffic patterns on the rack-level, and additional modifications should be done on every host and every switch.

The same recognition can be seen also in vehicle traffic system. As proposed in [20], NMF algorithm is used to realize OD matrix (very similar to the traffic matrix) decomposition. Though the NMF can recognize features of a matrix, the features are without physical significances. But these features can be used to predict future via AR algorithm.

C. RECONFIGURABLE TOPOLOGY IN OPTICAL DATA CENTER NETWORK

Since the optical interconnection network is introduced in data center, multiple switching modes can coexist in the network. Thus, by combination of the switching modes, the logical topologies (i.e. virtual topologies) are diversified into multiple basic types. Further, the topologies can be dynamically configured. For example, in Helios, which has been earliest proposed, the basic tree topology uses electrical packet switching (EPS), and the reconfigured light paths can be implemented via optical circuit switching (OCS). In the OpenScale network [11], the basic hexagon topology runs optical packet switching (OPS), and as well, the OCS is used to build the logical connections between nodes.

As for the topology reconstruction strategies, the primary one is periodical reconstruction [21]. It can obtain good performance when the network traffic distribution is even. And it is easily deployed without concerns on traffic patterns adaption. Then, the topology computations aiming at throughput maximization are proposed. In Helios, it implemented the topology reconstruction strategy, i.e. the Hedera, which is no difference with the *b-matching* algorithm. The *b-matching* algorithm regards the traffic demand as a bi-graph, the corresponding topology is just maximum matching of the bi-graph. And a TATR [11] method is proposed based on the OpenScale network. The TATR will preferentially bridge the racks with larger communication cost (i.e. the traffic volume multiplies the hops). While in [22], a latency-driven topology reconstruction method implemented via deep learning is proposed. The objective for this topology reconstruction is to minimize the flow completion time. In this work, the correlative of flows is not considered. Besides, the time complexity of this method is too high, because the topology searching is across whole network (i.e. global search) and several extra deep learning models have jointly been used.

D. THE COFLOW ACCELERATION

The Coflow acceleration can be achieved from several aspects. One is to properly place the tasks of a job so that the communication costs can be reduced, or the network congestion can be avoided. Methods like ShuffleWatcher [23], Corral [24] and SMD [25] are belong to this kind. The second one is to design the flow-level scheduling to achieve preemptively inter-Coflow forwarding, such as WSS [26] and SCF/SEBF [27]. The above optimizations may acquire prior knowledge from the application, such as the specific traffic volume inside a Coflow. The third one is to dynamically adjust the topology to match the traffic patterns of Coflow. The demonstrations have verified the feasibility in the "a table for two topologies" [9] and the "OvS" [10]. However, the specific strategy is not clear in those experiments.

III. TRAFFIC PATTERN RECOGNITION AND TOPOLOGY RECONSTRUCTION

The aim of our recognition is neither simple classification of size or type, nor the prediction via history data. Instead, the traffic patterns are expected to be recognized, and the on-demand topology can be provisioned accordingly. Based on the aforementioned analyzation, the traffic patterns are generated from several concurrent computing jobs. If the more accuracy recognition can be done to know the group of racks for a job, the more adaptive topology can be reconstructed to match the traffic pattern of this job. Since all of the traffic patterns of concurrent jobs are recognized, and the light paths of the network are reconfigured to adapt the concurrent jobs, these jobs can be accelerated simultaneously.

It can be noticed that to recognize the traffic patterns without prior knowledge is actually to cluster the racks with strong connections into a group. If the traffic requests among the racks are regarded as a weighted graph, the clustering of racks can be modeled as a weighted graph cutting problem. The weights in a graph indicates the traffic volume. To achieve the weighted graph cutting, the basic unsupervised clustering methods (i.e. the *KNN*, *k-means* and community algorithms) are less efficient. Because the number of the rack groups

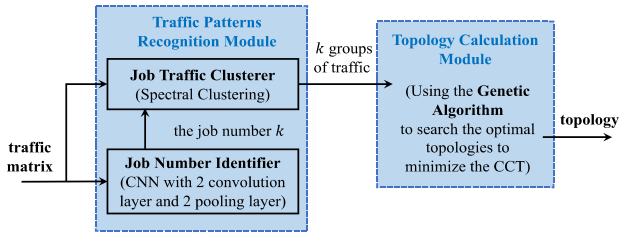

FIGURE 2. Procedure to implement traffic patterns -aware topology reconstruction.

may not be deterministic. Besides, the *KNN* or the *k-means* is more efficient to cluster the data sequence rather than a graph. The k-clique method in community algorithms can be used to cut the graph with multiple full-meshed subgraphs, which is inapplicable for our case. (Even the traffic patterns in a job are clustered and corelative, they are not full-meshed.) Although the other community method, the fast-unfolding theoretically can be used, its performances are not good, the analysis of which can be seen in Section 4. The spectral clustering algorithm [28] has also been verified to be suit for the graph cutting. Unlike the community algorithm, to use the spectral clustering, the number of the groups can be pre-defined. If such parameter is not previously given, the cluster number has to be tested by the clustering algorithm until the results are accuracy. The time complexity will increase. So, the cluster number is expected to be known at first. To know the group number of the graph can be modeled as a graph classification problem. The CNN can extract the graph features via convolutional processing, and then classify the graphs. In light of these, the traffic patterns recognition method is CNN combined with spectral clustering.

As for the topology reconstruction strategy, it will benefit for lowering the CCT, so that the JCT can be decreased. It is not easy to find a direct method to obtain a topology to minimize the CCT of a job. Thus, the problem can be transformed as an optimization model. The optimization goal is to minimize the CCT in a certain group of clustered traffic. To solve this optimization model, i.e. to search a topology which can obtain minimum CCT, the genetic algorithm is utilized for its fast convergence speed.

The procedure to realize the traffic patterns -aware topology reconstruction strategy is shown in FIGURE 2. In the traffic patterns recognition module, to recognize the traffic patterns is to identify the job number k in the traffic matrix via the job number identifier, and then to cluster the racks to k groups via the job traffic clusterer. The identification of the job number k is a classification problem. For example, the traffic matrix with $i(i = 1, 2, \ldots, k)$ jobs can be marked as the i class. A trained CNN is utilized to identify the job number k. After the k is determined, the traffic matrix will be clustered to k group by the stage-of-art spectral clustering algorithm. Then, the recognized k groups of traffic will be sent to the topology calculation module. After searching via genetic algorithm, a topology with minimal CCT can be obtained.

Algorithm 1 Spectral Clustering

Input: k, **TM** (TM: Traffic Matrix)
TM $\leftarrow$ Relative(**TM**)
if $i = j$ **then**
  $\mathbf{D}(i, j) = \mathbf{TM}(i, j)$
end if
$\mathbf{L} = \mathbf{D} - \mathbf{TM}$
$\mathbf{L} \leftarrow$ Normalized($\mathbf{L}$)
$\mathbf{EV} =$ Eigenvector($\mathbf{L}$)
$\mathbf{RG}_k =$ K-means($\mathbf{EV}, k$)
($\mathbf{RG}_k$: k rack groups)

A. TRAFFIC PATTERN RECOGNITION

We used CNN to classify the traffic matrix. The CNN contains several convolution layers and an equal number of pooling layers. As shown in FIGURE 3, the input graphs are convolved by a convolutional kernel function (i.e. a small square matrix) to multiple feature maps. Then a pooling function will subsample these feature maps to smaller size. Then these feature maps will be sent to next convolution layer and the corresponding pooling layer. After each convolution layer and pooling layer, the number of the feature maps increases, but the size of the feature maps is smaller. The final feature maps will be set relation to the output vector. The output vector is used to label the class of the traffic matrix. For example, we can use k-size hot key vector [29] to represent k class. When training the CNN, the convolutional kernel function will be upgraded according to the learning errors. After multiple learning steps, a best kernel function will be learned until the error is limited and stationary.

The trained CNN can be used to directly classify a traffic matrix. After classification, the traffic matrix and the k will be used as inputs of the spectral clustering model. The outputs of the spectral clustering are several groups of racks. A certain job is running on one of these rack groups. The spectral clustering algorithm clusters these groups according to the relationship between racks. Such relationship is actually described in the traffic matrix. For example, the element of i row and j column indicates the relationship of rack i and rack j. The relationship is strong if the traffic between i and j is large. Furthermore, the correlative relationship among multiple racks can also be captured by spectral clustering algorithm. The principle of the spectral clustering algorithm is that the input traffic matrix will be transformed to a *Laplace* matrix. And the eigenvectors of the Laplace matrix will be calculated. Then, these eigenvectors will be clustered by the deterministic *k-means*. The number of the clustering group is according to the input k. The pseudo-code of the spectral clustering is shown in "Algorithm 1". Depending on CNN and spectral clustering, the traffic patterns are recognized completely.

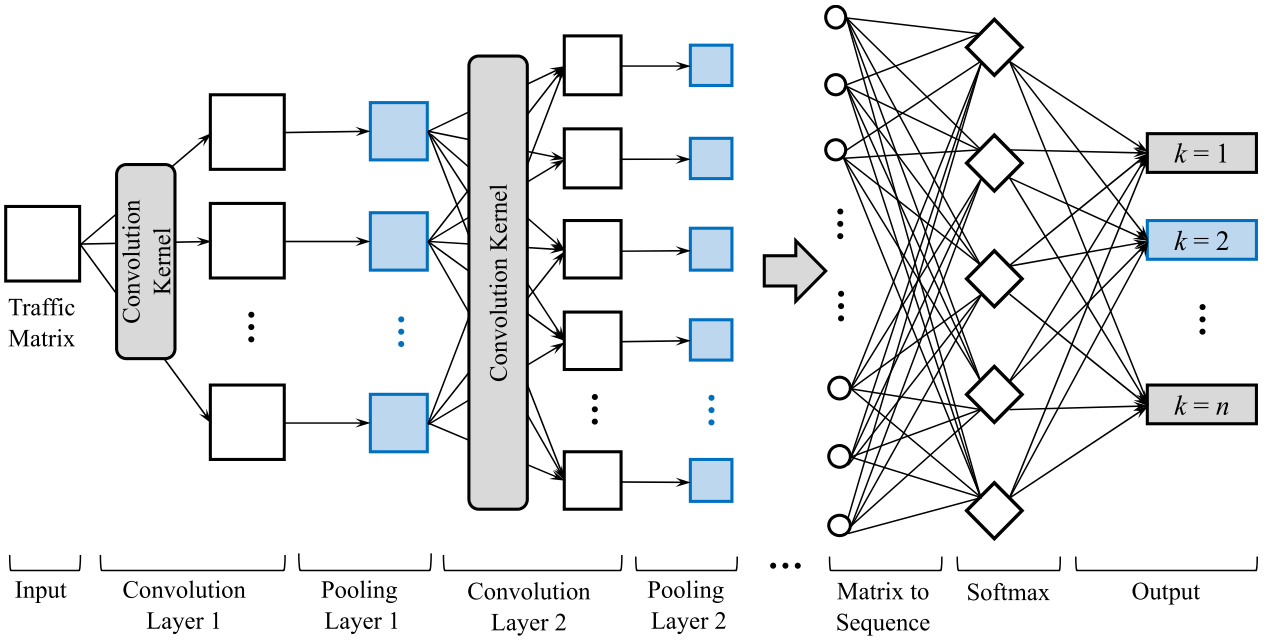

FIGURE 3. CNN structure to classify the clustered traffic patterns.

B. TOPOLOGY RECONSTRUCTION STRATEGY

The adaptive topology ready to be reconstructed will be searched through genetic algorithm. The objective of the searching is to find a topology which can minimize the CCT. The CCT calculation cannot directly be represented as an equation. It can be calculated by mapping traffic matrix to a certain topology. Using specific routing (i.e. shortest path in this paper), the traffic flow is assigned to each network node. And using single queue model, the process latency of traffic flow in each node can be calculated. Then, one flow completion time equals to the sum of the process latencies of routed nodes of this flow. Finally, the CCT is the maximum flow completion time in the same job.

The topology searching will be operated to each rack group. Before the genetic algorithm, each rack ID will be encoded to a binary vector. A light path can be represented as a joint vector of two different binary vector of racks. In the genetic algorithm, *pn* initial topologies are randomly generated as the initial genes. And then based on *pn* initial topologies, the CCT is calculated, and the fitness is assigned inversely proportional to the CCT. And in the subsequent iteration, the Roulette Wheel Selection (RWS) is used. It means a gene (i.e. a topology) will be selected obeying the selection probability. The selection probability is calculated according to the fitness. But in order to accelerate the conversion, we set the selection probability of the topology with smallest CCT to 1. It means that such topology will appear in next generation definitely. And the one-point crossover and the simple mutation will be used in every iteration. After multiple iterations, the genetic algorithm will stop if the minimum CCT is stationary or due to the maximum step (i.e. the end condition). Noteworthy, in every iteration, the CCTs are calculated based on the shortest path routing. The proper topology may vary when the routing method is changed. The details of the genetic algorithm are shown in the following pseudo-code "Algorithm 2".

Algorithm 2 Topology Searching

```
Encode()
TS = Initialize(pn) (TS: Topology Set)
while True do
    CCT ← CalculateCCT(TS)
    Fitness ∝ 1/CCT
    Probability(i) ← Fitness(i)/Sum(∑_{i=1}^{pn} Fitness(i))
    Maximum(Probability) ← 1
    TS ← Crossover(TS, Probability)
    TS ← Multation(TS, Probability)
    TS ← Insert() (insert a random topology)
    (next generation of topology)
    EndCondition() (break or continue)
end while
```

IV. SIMULATION EVALUATION

We evaluated the performance of traffic pattern-aware topology reconstruction strategy by simulation. Specifically, the classification accuracy of CNN, the clustering accuracy of spectral clustering algorithm, and the CCT promotion rate of such strategy have been analyzed. The simulation is programmed by Python and ran on a VM from Dell 720 server with 4 CPU cores and 16GB memory.

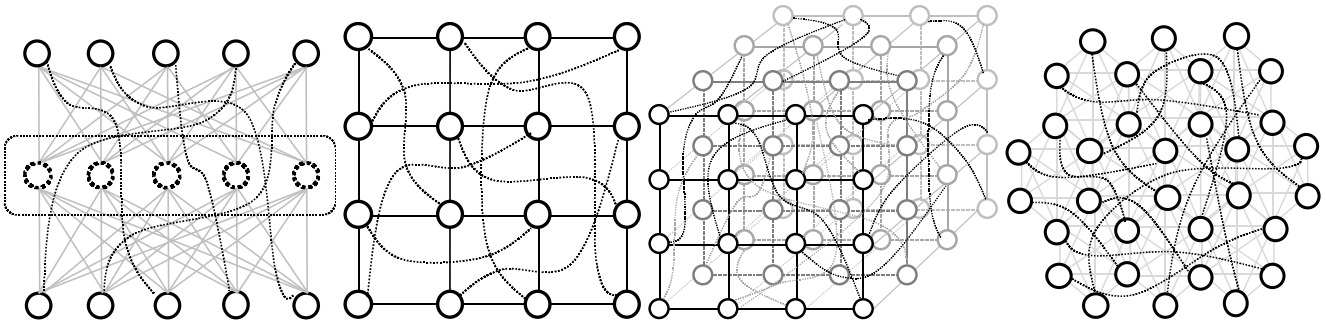

FIGURE 4. Four networks to evaluate the traffic patterns-aware topology reconstruction.

A. SETUPS

Four networks are used to evaluate the performance of the traffic pattern -aware topology. The four networks are drawn in FIGURE 4. These networks basically have a fixed topology (e.g. through electrical packet switching (EPS), or optical packet switching (OPS)). And each node (i.e. rack-level) has d ports (e.g. in the FIGURE 4, the $d = 1$ for more visible view, also in the evaluation) to interconnect the other nodes to build the reconfigurable light paths. The fixed topology of the four networks are: Tree, Lattice, Cubic and Small-world. In the Tree network, all the nodes can be seen to be connected with a big packet switch. And in the Lattice or Cubic network, nodes can be fixedly connected as multiple squares or cubes. While in the Small-world network, fixed topology can be regarded as multiple full-meshed hexagon cells.

In the simulation, the number of the nodes of all the networks is 216. So that the traffic matrix is 216×216. Each traffic matrix may contain k jobs, the maximum k is 6. 1200 traffic matrixes are generated according to a real job trace from Facebook [14]. In the traffic pattern recognition, these traffic matrixes will be transformed into relative ones (i.e. each element in the matrix is no more than 1) to avoid the potential effect due to the uneven traffic volume.

The CNN is set with two convolutional layers and two pooling layers. The input of the CNN is a relative 216×216 traffic requests matrix. In the first convolutional layer, the convolutional kernel is 3×3 and the outputs are 32 150×150 feature maps. And these feature maps are reshaped to by the first 3×3 pooling layer to 72×72. Then the 32 72×72 maps are input to the second convolutional layer; the outputs are 64 72×72 feature maps. After the second 3×3 pooling layer, the 64 feature maps are reshaped to 24×24. The output layer uses the 64 feature maps to map a vector with 6 elements. In the spectral clustering, the *k-means* is used as the eigenvalue clustering method. In the genetic algorithm, the population number pn=10.

1000 pairs of traffic matrix and the previously known k are used to train the CNN. And the residual 200 traffic matrixes are used to test the trained CNN. Then, based on these 200 traffic matrixes, the job traffic clustering and topology reconstruction have been evaluated.

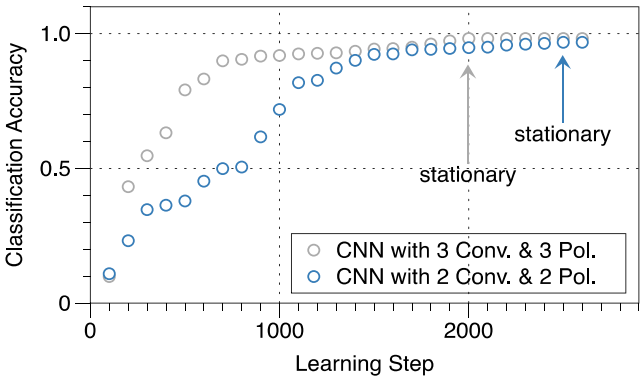

FIGURE 5. The classification results of CNN.

B. TRAFFIC RECOGNITION ANALYZATION

The classification accuracy of the CNN is evaluated. In the evaluation, we compare the performance of two CNNs, one is with 3 convolutional layers and 3 pooling layers (depicted as blue spot in FIGURE 5), the other is with 2 convolutional layers and 2 pooling layers (depicted as grey spot in FIGURE 5). It can be seen that the CNN with more layers can achieve fast convergence along learning steps. However, the training time of each learning step is much longer than the CNN with less layers. Though the CNN with less layers converges slowly and shows less classification accuracy, the training speed is fast. So, we used the CNN with 2 convolutional layers and 2 pooling layers as aforementioned, because the accuracy deterioration is not evident. The following analyzation is based on this.

In the training process, the classification accuracy of the traffic matrix can arrive to 96.8%, and when testing, the accuracy can keep at 88.1%. And based on the 200 traffic matrixes, the accuracy of traffic pattern clustering is also evaluated in FIGURE 6 (b). This accuracy in spectral clustering is defined as:

$$\frac{1}{k}\sum_{i=1}^{k}(1-\frac{E_i}{N_i})$$

In clustered group i, the E_i is the number of the missing and wrong nodes, and the N_i is the actual number of nodes. In the FIGURE 6 (b), it can be observed that, when the k increases,

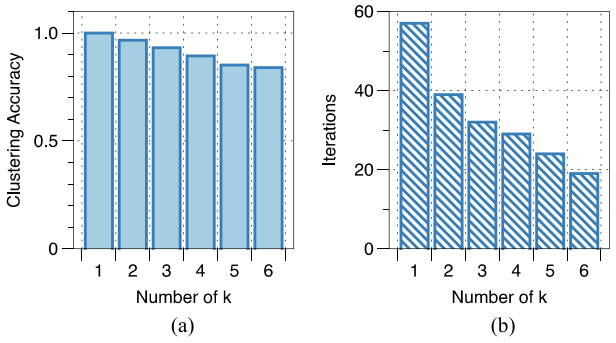

FIGURE 6. (a) The average clustering accuracy by spectral clustering algorithm. (b) The average iteration step by genetic searching algorithm.

the clustering accuracy drops. This is because the k groups may overlap. But a certain node can only be clustered to only a group, so the overlapped nodes may be missing in other groups.

We compared the performances of spectral clustering and the fast-unfolding algorithm. When using the fast-unfolding algorithm, the average accuracy can reach almost 0. This is because the k is always falsely recognized by the fast-unfolding method. It proves that the community algorithm cannot be used in our cases.

We also analyzed some key performance indexes of spectral clustering, including root mean square standard deviation (RMSSTD), the R-square (RS) and the calinski_harabaz index (CH). The smaller RMSSTD indicates more accurate clustering. The RS is a number between 0 and 1. When the RS is closer to 1, it means the clustering results are more approximate to real data. And the CH index can be used to compare the inter-group distance and intra-group distance. The higher CH means the larger inter-group distance and smaller intra-group distance, and ultimately, better clustering performances. The average RMSSTDs under different k are depicted in FIGURE 6. (a). The average RSs of k are shown in FIGURE 6. (b). In FIGURE 6. (c), it can be seen that the CHs get highest when the k is accurately given. This can prove the efficiency of our hybrid methods.

C. TOPOLOGY RECONSTRUCTION ANALYZATION

In the topology calculation, the convergence rate of the genetic algorithm is evaluated in FIGURE 6 (b). The number of convergence step decreases by growth of k. Because the bigger k indicates smaller number of nodes in a group. The number of light paths is limited, so the convergence step is lower.

And the CCT promotions based on calculated topology (via the traffic patterns -aware reconstruction, the TATR reconstruction and periodical reconstruction) are compared to the traditional *b-matching* reconstruction in FIGURE 8. The promotion is using CCT based on *b-matching* divided by the CCT based on other topology reconstruction methods.

Under the traffic patterns -aware reconstruction, the CCT promotions rise when the k is larger. The reason is that when the network scale increases, the communication costs within a small group of nodes have larger probability to be higher. This is because the network distance (i.e. hops) between nodes may be longer. In this case, the traffic patterns -aware reconstruction strategy can recognize the traffic correlation in large-scale and revise the mismatch between topology and traffic. So, the topology strategy for the clustered traffic can give better optimization on larger k clustered traffic.

Under the TATR, the CCT promotion is lower when k is smaller. And when the k grows, the CCT promotion of TATR gets better than *b-matching*. Because the TATR can partly restrict its reconstructions in a certain cluster. But the TATR does not aim to accelerate the CCT, thus, the promotion is limited.

Oppositely, the periodical reconstruction may strongly impact the flatten network (i.e. the small world, the cubic and the lattice network). And when the k increases, the objectless periodical reconstruction can cause huge mismatch between traffic and topology, and ultimately the performance of periodical reconstruction gets worse.

D. TIME COMPLEXITY ANALYSIS

The time complexity of CNN is $O(F^2 \cdot K^2 \cdot C_i \cdot C_o)$. The F is the maximum size of feature maps, the K is the maximum of convolution kernel. And the C_i/C_o is the input/output dimension. In this paper, the F is 72, the K is 3, and the C_i/C_o is 1. And the time complexity of the spectral clustering method is determined by the method for computing the eigenvalues. It could be $O(n^2)$ or $O(n^3)$. In this paper, we choose method with time complexity of $O(n^3)$. The community clustering method's time complexity is $O(mn)$, in which n is the vertex size, and the m is the edge size. When the graph is denser, the $O(m)$ can be closer to $O(n^2)$. (When the graph is full-meshed, the m=n(n−1)/2, thus $O(m)$ is $O(n^2)$.) In this paper, the traffic matrix is dense. Thus, if the community method is used, the complexity will arrive $O(n^3)$.

As for the topology searching, the time complexity is mainly impacted by the iteration steps, namely nT. The n is the step number of iterations, and the T is the average time for each iteration. In [22], it uses global searching, the n and the T are both higher. However, in this paper, the topology is only searched in each cluster. Thus, the n and the T is much smaller.

Besides the above theoretical analysis, we also evaluated the real time of clustering and topology searching in Section 6.

E. ADVANCED DISCUSSION

How does the classification error in CNN impact the clustering, and then the CCT promotion? If the deviations of is larger in classification, the accuracy is worse in clustering. In order to detail this phenomenon, as shown in FIGURE 9, a clustering (i.e. the real k is 3) under k from 2 to 6 is evaluated. It can be seen that when the accuracy goes worth when the k is 2 or 6. Based on the wrong clustering, we also evaluate the corresponding CCT. When the k is too larger

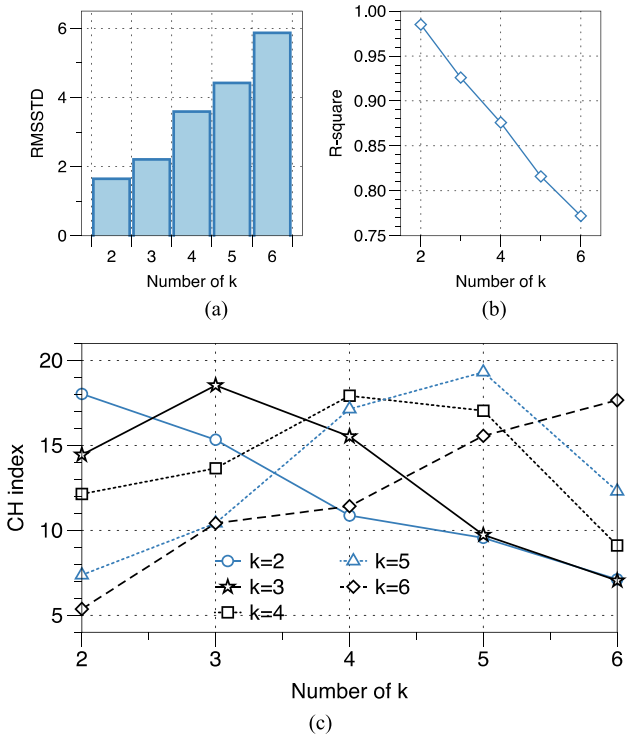

FIGURE 7. (a) The RMSSTD of spectral clustering algorithm. (b) The R-square of spectral clustering algorithm. (c) The CH index of spectral clustering algorithm.

against the real, the CCT promotion is much worse. This may because the over-clustering breaks the correlative of the traffic. And it can be also noticed that comparing the four networks, the Tree topology which gets highest promotion when k is accurate shows severer deterioration when k is incorrect. To illustrate the reason for this case, we can trace back to the fixed topology of each network. Each fixed topology shows different performance originally. For example, in the network with Lattice topology, the CCT is larger because the traffic may suffer more hops (e.g. the Lattice network). While in the network with Tree topology, the CCT increases due to more probability of congestion. But the Small-World fixed topology naturally has shorter average network distance, and the CCT may be lower than other topologies. When the traffic patterns are accurately recognized, and the topology is properly reconstructed, the CCT can be accelerated more (e.g. though the absolute CCT on Tree is lower than the CCT based on Small-World, the CCT promotion is higher) because the reconstructed topology can gloss over the natural defects of fixed topology. On the contrary, when the recognition is bad, and the topology mismatches, the CCT may be directly influenced by the nature of fixed topology.

But how worse the k is when the classification goes wrong in CNN. It has also been analyzed. When the is mistakenly classified, the k is not far from the real one, namely just close to it (e.g. the real k is 3, the wrong k may be 2 or 4 under large probability). So, even the CNN is not accurate, but the final CCT promotion may not be impacted a lot.

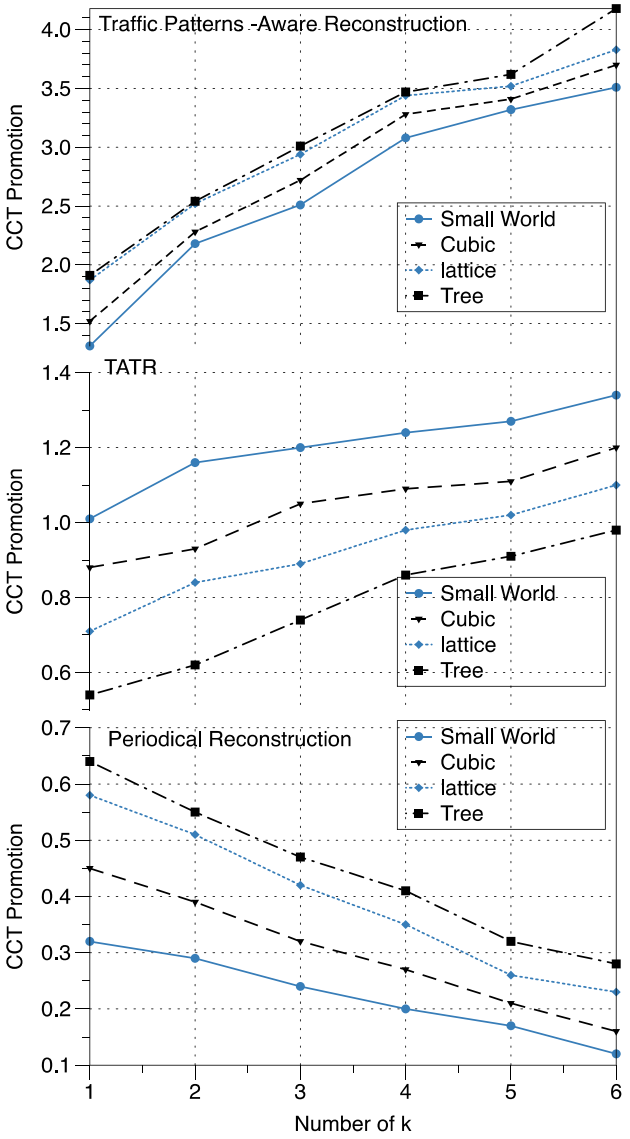

FIGURE 8. The average CCT promotion after traffic patterns-aware topology reconstruction, TATR topology reconstruction and periodical topology reconstruction (using 200 test traffic matrixes).

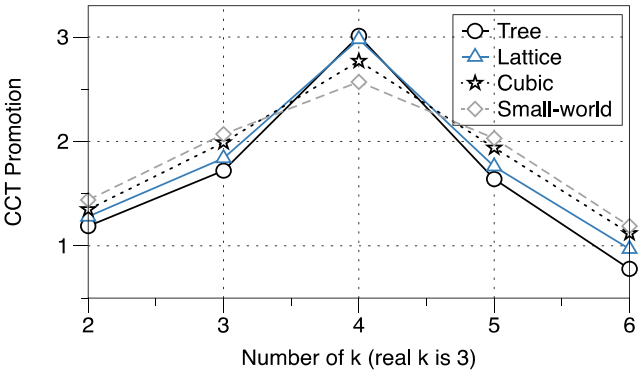

FIGURE 9. The average CCT promotion deterioration caused by classification error.

How does the clustering error in spectral clustering algorithm impact the CCT promotion? When recalling the definition of the clustering accuracy, the wrong, missing

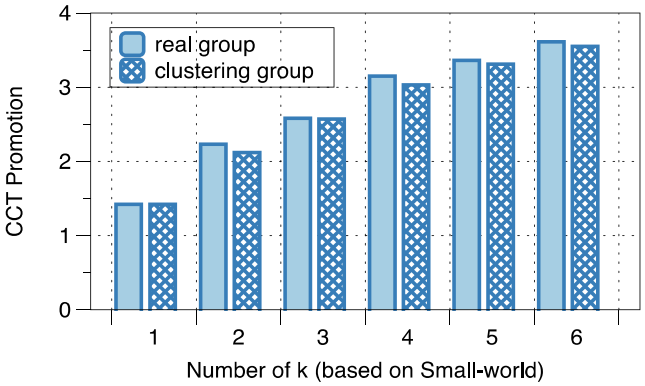

FIGURE 10. The average CCT promotion deterioration caused by clustering error.

nodes can increase the error rate. Actually, based on our analyzation, the wrong nodes are really rare in the spectral clustering, the missing nodes contribute more to the error rate. And the missing nodes are caused by the overlap of the jobs. But what is the incentive in the spectral clustering that judges the overlap nodes into a certain group? According to the principle of the spectral clustering, when the traffic volume between two nodes is larger, the two nodes may be clustered together more probably. So, the missing nodes may have weaker connections to a first group, but have stronger connections to a second group. As a result, these missing nodes will be clustered to the second group, and then the accuracy of the first group decreases.

To verify the influence of the missing nodes, the CCTs have been evaluated under the real group or under the group from the spectral clustering, as shown in FIGURE 10. The deterioration is not evident. It may be because that the heavy traffic has been clustered. And compared to the heavy traffic, such missing nodes with small traffic demands may not strongly impact the CCTs.

How far is the difference between the clustered traffic aware topology reconstruction and the b-matching? It has been compared by the monochrome pictures, as shown in FIGURE 11. Ranging in partial network nodes (i.e. from 100 to 200), only light paths for one group of traffic are depicted for more visualization. The black pixel represents a light path. The FIGURE 11 (a) shows the light path distribution under b-matching, while the FIGURE 11 (b) shows the light path distribution under traffic patterns -aware strategy. The distribution of the light paths under two reconstruction strategies are much difference due to very little coincidence of the black pixels. We cannot verify to enlarge the throughput and to accelerate the CCT are contradictions. But they are really two optimization directions.

How about the small Coflow? Someone may argue that small Coflows are relatively small. How these Coflow can be recognized. Actually, our work is to treat the Coflow which has inter-rack communications. So, the relatively small Coflow is not considered. But if the ignorance will deteriorate the CCT? Because the jobs with small Coflow may be running

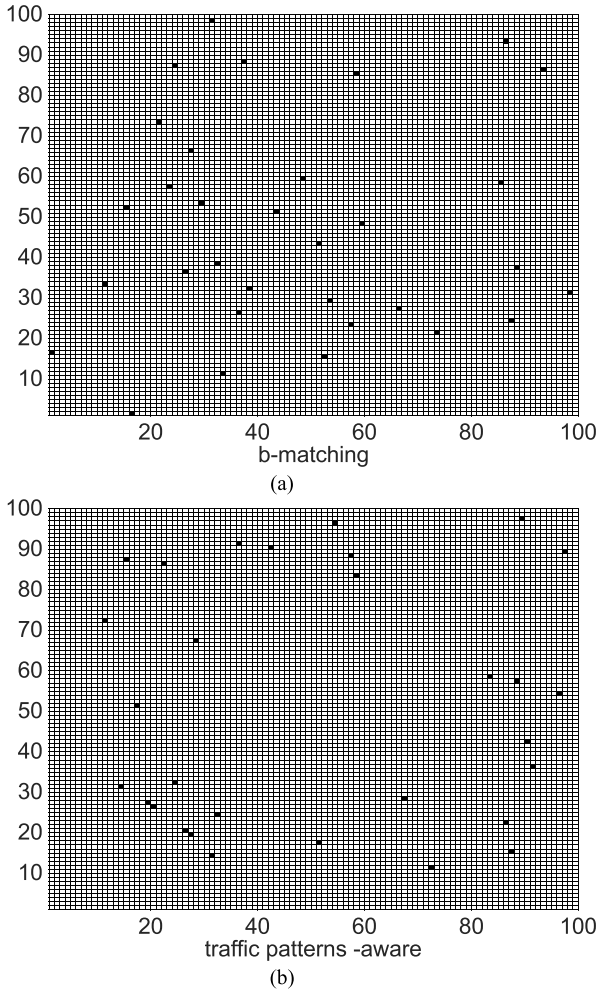

FIGURE 11. Light path distribution based on (a) b-matching; (b) traffic patterns-aware topology reconstruction (a black pixel represents a light path).

within a rack, so that the inter-rack communications are not required, and the ignorance may not impact them.

V. EXPERIMENTAL DEMONSTRATION

Then the topology reconstruction strategy is also demonstrated in an experimental way. In the experiment, 16 nodes were originally connected as a Tree topology and then the Small-world topology. Each node had a port to build a light path. The light path could be reconfigured via control of the fast optical switching matrix (∼300ns). Under each switching node, two VMs were started up to run the computing jobs. The Spark was used as the computing framework. The HiBench [30] was deployed as the computing job benchmark. Two types of jobs, the *Sort* and the *WordCount* were mixed to run on the network to generate 20 clustered traffic patterns (the maximum k is 4). The traffic matrix of each traffic pattern has been known via previous measure of the job communications. Based on recognition of these traffic pattern, the topology strategies were calculated.

We firstly tested the real processing time of clustering and topology searching. In FIGURE 12. (a), it can be observed

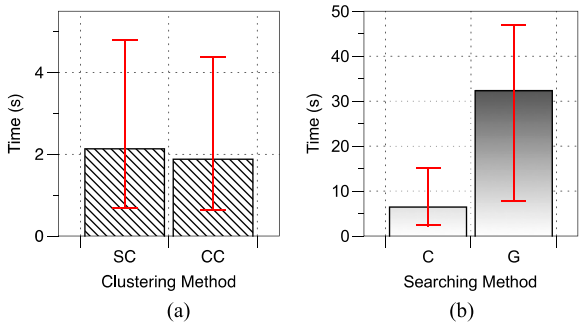

FIGURE 12. (a) The real computation time of clustering method (SC: spectral clustering; CC: community clustering). (b) The real computation time of topology searching; (C: our clustered approach; G: global approach in [22]).

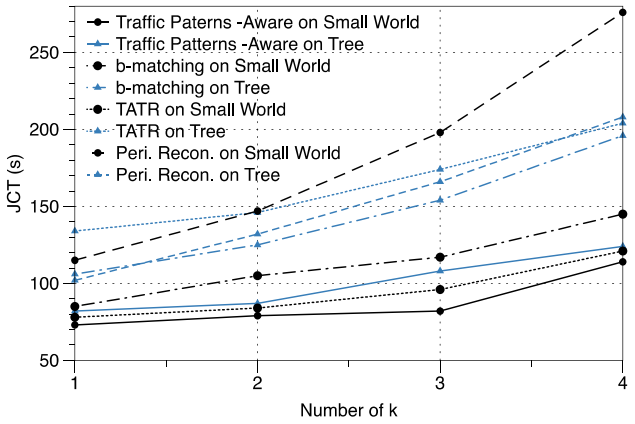

FIGURE 13. The JCT evaluation for different number of k (ranges from 1 to 4).

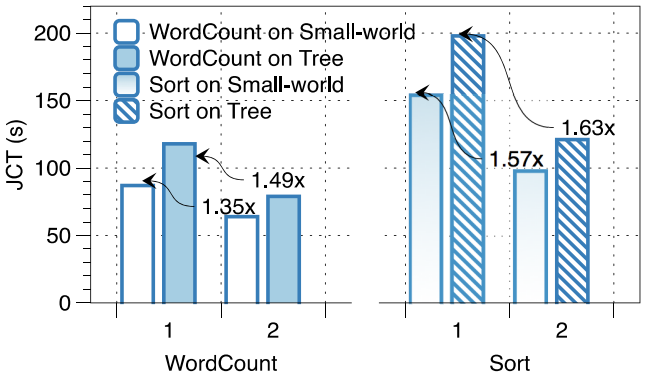

FIGURE 14. The JCT evaluation for different types of job (1: the traffic patterns -aware topology; 2: b-matching topology; 3: TATR; 4: Periodically reconstruction).

that the processing time of spectral clustering is a little higher than community clustering. However, the latter method suffers low accuracy rate. The processing time of global searching method in [22] and our clustered approach are shown in FIGURE 12. (b). The processing time of global searching is much higher than ours. The efficiency of our work can be verified then.

Then we evaluated the average JCT of different k, under periodical strategy, TATR strategy, *b-matching* strategy and

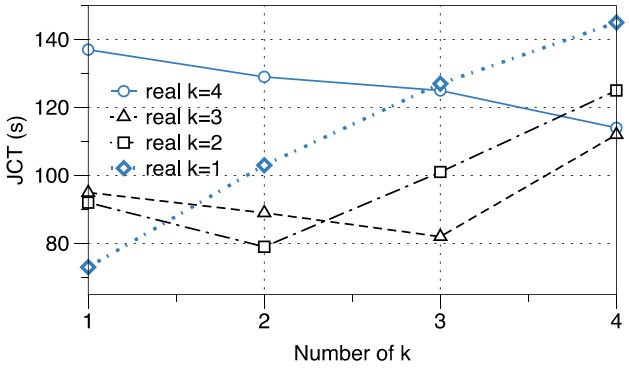

FIGURE 15. The JCT evaluation when the parameter k varies.

the traffic patterns -aware strategy. In FIGURE 13, it can be confirmed that the absolute JCT on the network with fixed Small-world topology is lower, but the JCT can be accelerated more on the network with fixed Tree topology.

The average JCTs of *Sort* and *WordCount* are also compared. In FIGURE 14, it can be seen that the *Sort* job can obtain better performance. This may be because the *Sort* is the network-intensive job [10]; the traffic requests from *Sort* job is larger than *WordCount*. So, if the communication costs of *Sort* job are reduced by the topology reconstruction, the JCT of this job can be much lowered.

In the experiment, the parameter that may influence the JCT is the clustering result. Follow the simulation if FIGURE 9, we analyzed the JCTs when the k varies on the small world network. In the FIGURE 15, it can be found that the JCT is lowest when the k is correct. And the k is less accurate, the JCT will be longer.

VI. CONCLUSION

To fully explored the topology flexibility of optical network to optimize the computing application in DCs, this paper proposed the traffic patterns -aware topology reconstruction strategy. To face the challenge of traffic patterns recognition, the CNN plus spectral clustering model is utilized to recognize the traffic patterns without prior knowledge from application. Based on the recognized patterns, the topology strategy for minimizing the CCT via genetic algorithm has been verified the good performance through both simulation and experiment way. Therefore, the performance of computing system can be promoted. Although the evaluation is based on a few kinds of networks, such strategy can be utilized in any other optical architecture which is capable for topology reconstruction.

On the other side, in some cases, especially when the jobs are overlapped for larger scale, our approaches suffer performance deteriorations. However, such cases are less. In most cases, our methods can efficiently optimize the computing jobs.

REFERENCES

[1] *Hadoop Website*. Accessed: Mar. 11, 2018. [Online]. Available: http://hadoop.apache.org

[2] *Spark Website*. Accessed: Mar. 11, 2018. [Online]. Available: http://spark.apache.org
[3] C. Mosharaf and I. Stoica, "Coflow: A networking abstraction for cluster applications," in *Proc. ACM Workshop Hot Topics Netw.*, 2012, pp. 31–36.
[4] F. R. Dogar, T. Karagiannis, H. Ballani, and A. Rowstron, "Decentralized task-aware scheduling for data center networks," in *Proc. ACM Conf. SIGCOMM*, 2013, pp. 431–442.
[5] S. Ye, Y. Shen, and S. Panwar, "HELIOS: A high energy-efficiency locally-scheduled input-queued optical switch," in *Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst.*, 2010, p. 8.
[6] G. Wang *et al.*, "c-Through: Part-time optics in data centers," in *Proc. ACM Conf. SIGCOMM*, 2010, pp. 327–338.
[7] K. Chen *et al.*, "OSA: An optical switching architecture for data center networks with unprecedented flexibility," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 498–511, Apr. 2014.
[8] J. Edmonds, "Paths, trees and flowers," *Can. J. Math.*, vol. 17, pp. 449–467, 1965.
[9] Y. Xia, X. S. Sun, S. Dzinamarira, D. Wu, X. S. Huang, and T. S. Ng, "A tale of two topologies: Exploring convertible data center network architectures with flat-tree," in *Proc. ACM Conf. SIGCOMM*, 2017, pp. 295–308.
[10] Z. Zhu, S. Zhong, L. Chen, and K. Chen, "Fully programmable and scalable optical switching fabric for petabyte data center," *Opt. Express*, vol. 23, no. 3, pp. 3563–3580, 2015.
[11] D. Zhang, J. Wu, H. Guo, and R. Hui, "Optical switching based small-world data center network," *Comput. Commun.*, vol. 103, pp. 153–164, May 2017.
[12] D. Wei, L. Xu, X. Jin, Y. Li, and W. Xu, "A 12-rack, 180-server datacenter network (DCN) using multiwavelength optical switching and full stack optimization," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2016, pp. 1–3, Paper Th5B.6.
[13] D. Zhang, H. Guo, G. Chen, Y. Zhu, H. Yu, and J. Wang, "Analysis and experimental demonstration of an optical switching enabled scalable data center network architecture," *Opt. Switching Netw.*, vol. 23, pp. 205–214, Jan. 2017.
[14] *Coflow Benchmark*. Accessed: Apr. 19, 2018. [Online]. Available: https://github.com/coflow/coflow-benchmark
[15] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron, "Better never than late: Meeting deadlines in datacenter networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 50–61, Aug. 2011.
[16] M. Channegowda, T. Vlachogiannis, R. Nejabati, and D. Simeonidou, "Optical flyways for handling elephant flows to improve big data performance in SDN enabled datacenters," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2016, pp. 1–3, Paper W3F.2.
[17] L. Chen, K. Chen, W. Bai, and M. Alizadeh, "Scheduling mix-flows in commodity datacenters with karuna," in *Proc. ACM Conf. SIGCOMM*, 2016, pp. 174–187.
[18] N. Fernández *et al.*, "Virtual topology reconfiguration in optical networks by means of cognition: Evaluation and experimental validation," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 7, no. 1, pp. A162–A173, Jan. 2015.
[19] H. Zhang, L. Chen, B. Yi, K. Chen, M. Chowdhury, and Y. Geng, "CODA: Toward automatically identifying and scheduling coflows in the dark," in *Proc. ACM Conf. SIGCOMM*, 2016, pp. 160–173.
[20] X. Li, J. Kurths, C. Gao, J. Zhang, Z. Wang, and Z. Zhang, "A hybrid algorithm for estimating origin-destination flows," *IEEE Access*, vol. 6, pp. 677–687, 2018.
[21] W. M. Mellette *et al.*, "RotorNet: A scalable, low-complexity, optical datacenter network," in *Proc. ACM Conf. SIGCOMM*, 2017, pp. 267–280.
[22] M. Wang *et al.*, "Neural network meets DCN: Traffic-driven topology adaptation with deep learning," in *Proc. Abstr. ACM Int. Conf.*, 2019, pp. 97–99.
[23] F. Ahmad, S. T. Chakradhar, A. Raghunathan, and T. N. Vijaykumar, "Shufflewatcher: Shuffle-aware scheduling in multi-tenant mapreduce clusters," in *Proc. USENIX ATC*, 2014, pp. 1–13.
[24] V. Jalaparti, P. Bodik, I. Menache, S. Rao, K. Makarychev, and M. Caesar, "Network-aware scheduling for data-parallel jobs: Plan when you can," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 407–420, 2015.
[25] C. Wang, H. Guo, D. Zhang, and J. Wu, "Topology-aware task placement in small-world optical data center network," in *Proc. Opto-Electron. Commun. Conf. (OECC) Photon. Global Conf. (PGC)*, Jul. 2017, pp. 1–3.
[26] M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, "Managing data transfers in computer clusters with orchestra," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 98–109, Aug. 2011.
[27] C. Mosharaf, Y. Zhong, and I. Stoica, "Efficient coflow scheduling with varys," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 443–454.
[28] F. R. Bach and M. I. Jordan, "Learning spectral clustering," in *Proc. Neural Inf. Process. Syst.*, vol. 16, no. 2, 2006, pp. 1–8.
[29] Z. Yan, V. Jagadeesh, D. Decoste, W. Di, and R. Piramuthu. (2014). "HD-CNN: Hierarchical deep convolutional neural network for image classification." [Online]. Available: https://arxiv.org/abs/1410.0736v1
[30] S. Huang, J. Huang, J. Dai, T. Xie, and B. Huang, "The HiBench benchmark suite: Characterization of the MapReduce-based data analysis," in *Proc. Int. Conf. Data Eng. Workshops*, 2010, pp. 41–51.

Authors' photographs and biographies not available at the time of publication.