HCIS | Human-centric Computing and Information Sciences

RESEARCH(Research Manuscript) Open Access

A Novel Load Balancing Scheme based on PFC Prediction in Lossless Datacenter Networks

Jin Wang[1], Yi He[1], Wangqing Luo[1] , Shuying Rao[1], Jinbin Hu[1,*]

Abstract

Response latency are critical for human-computer interaction experience. In recent years, the emerging datacenter applications require ultra-low latency and high throughput. The communication phase in those applications usually involves the collection of parallel flows, which is abstracted into the concept of coflow. Although the existing load balancing and coflow scheduling mechanisms achieve good performance in lossy datacenter networks (DCNs), they cannot work well in the lossless DCN employed with hop-by-hop Priority-based flow Control (PFC) technology. Because the existing load balancing mechanisms are obvious to the negative impact of PFC on the link status when making load balancing decisions for coflows, resulting in stalled flows. That means if a certain flow in a coflow encounters PFC pausing, the task will be stagnated, leading to serious performance degradation for the coflow's completion time (CCT), response latency and poor human-computer interaction experience. In this paper, we propose a novel user-centered Coflow Load Balancing (CLB) scheme based on PFC prediction in lossless DCN. CLB predicts PFC pausing through the derivative of queue length and considers the coflow characteristics when making load balancing decisions. The lightweight CLB can be flexibly and widely deployed in lossless DCN with low overhead. We conduct NS-3 simulation experiments under typical realistic datacenter workloads. Extensive simulations show that, comparing with the existing state-of-the-art load balancing schemes and coflow scheduling solutions, our scheme can achieve promising performance in reducing average coflow's completion time (ACCT), enhancing the human-centric interaction experience, and other performance metrics.

Keywords

Human-computer interaction experience; Response latency; User-Centered; Lossless Datacenter Networks; Coflow; Load Balance; PFC

1. Introduction

Nowadays, the emerging various datacenter applications such as MapReduce and user-oriented search platforms demand ultra-low latency and high throughput, which are committed to improve the high-quality human-computer interaction experience [1, 2, 3]. In practice, the response latency is critical for the users' sense of experience [4,5,6]. Modern DCNs adopt divide and conquer algorithm based on a tree topology [7, 8] to collect and feedback the corresponding information. They distribute computing and data on thousands of servers, which can greatly reduce the workload of each single server and enable users to obtain faster response time [9]. In this way, the response time is shortened and the human-centric interaction experience is significantly improved.

***Corresponding Author:** Jinbin Hu (jinbinhu@csust.edu.cn)

1 School of Computer and Communication Engineering, Changsha University of Science and Technology,ChangSha 410114,China

However, the work results of each node need to be transmitted, collected and processed before returning to users. Their communications are usually structured and occur between server nodes in successive computing phases. Generally speaking, the communication phase can not end until all the related flows are completed, so optimizing the coflow's [10] completion time (CCT) can reduce the job completion time accordingly [11, 12, 13]. In the process of processing these cluster computing applications in the DCN, if a certain flow or a certain working node processes the job slowly [14], users must wait for the data of this working node before proceeding to the next job [15]. However, this situation will lead to the phenomenon of disconnection, which will cause the completion time of coflow and the completion time of tasks to lag seriously, and ultimately make human-computer interaction experience very poor. Therefore, CCT is an important metric for the human-computer interaction experience.

Modern lossless DCN are based on low delay and lossless transmission, which usually use PFC technology [16, 17 , 18] to prevent buffer overflow and ensure lossless transmission. As the most widely used flow control technology that can effectively avoid packets loss, PFC is widely deployed in the fusion enhanced Ethernet of DCN as the basis of intelligent lossless networks. If the PFC function is used in the queue, we call it a lossless queue. When the lossless queue of downstream devices are congested, the downstream devices will notify the upstream devices to stop sending the traffic of the corresponding queue, and the PFC will suspend the relevant upstream ports. And when the buffer occupancy rate decreases to the PFC pausing threshold, the queue will resume transmission [16, 19], thus realizing zero packet loss transmission.

At the same time, in the DCN, each server node provides multiple parallel paths to achieve load balancing between server nodes and improve the throughput. However, given that network cards with small storage [20] pace are commonly used in data center networks, the lossless DCN adopt a simple Go-back-N retransmission scheme to deal with the problem of packet reordering. However, Packets transmitted through multiple paths with different delays may reach the receivers out of order, and the receivers' netcard will discard out-of-order packets. In addition, the senders will be required to retransmit all packets after the last acknowledgement packet, which greatly increases the CCT.

In recent years, many load balancing schemes [21, 22, 23, 24, 25] and coflow scheduling schemes [11, 12, 26, 27] have been proposed, these methods can effectively alleviate the links congestion problem and optimize ACCT. Although these load balancing schemes and scheduling schemes can perform well in lossy networks, they all have disadvantages in the lossless DCN. In the lossless DCN, existing load balancing mechanisms such as Hermes [23], LetFlow [22], CONGA [21], and coflow scheduling methods such as Varys [11] all do not consider hop-by-hop pauses caused by PFC. Therefore, when rerouting is performed in the DCN supported by PFC, it is likely that in a communication phase, a flow in the coflow frequently experiences PFC pausing in the transmission links, and the flow becomes a stagnant flow. Because of this, the average completion time of coflow is greatly increased.

Most existing load balancing schemes use local queue length, explicit congestion notification (ECN) [28], round-trip delay (RTT) [29], and link utilization as load balancing indicators to split flows on parallel paths. In addition, the links that have experienced PFC pausing might has a low load, which is likely to be mistaken as a non-congested path by the existing load balancing mechanism. As a result, more flows are allocated on this links, which further increases ACCT, the task will be blocked, and the performance will be degraded.

In view of the above low performance problems, we raised the following question: can we design a new lightweight and low overhead load balancing mechanism to predict the links that may experience PFC pausing. If we can, we will avoid such links and reduce the probability of coflow experience PFC pausing in the communication process, so as to avoid the frequent occurrence of stalled flows. At the same time, we can also consider the transmission characteristics of coflow to improve the average completion time of coflow. In this paper, we propose Coflow Load Balancing with PFC prediction (CLB) to answer this question affirmatively.

Specifically, CLB calculates the derivative of the egress queue size and preset a threshold below the PFC pausing threshold. Based on these three parameters, CLB classify switch's ports into four types of ports. The optimal path is selected for routing and forwarding to reduce the ratio of stalled flows caused by PFC pausing for coflows, result in reducing ACCT.

Our contributions can be summarized as follows:

1. We provide an extensive study to exploit why existing load balancing schemes in the lossless DCN with PFC mechanisms lead to flows stalling. Therefore, in order to effectively reduce CCT, we direct flows over the links that low probability of experencing PFC pausing.

2. We propose a user-centered load balancing scheme CLB, which helps the flows to keep away from the links that may experience PFC pausing with high probability when routing and forwarding on switches. The focus of our proposed scheme is its broad applicability and effectiveness in lossless DCN as well as its lightweight. It can significantly reduce response latency and improve the user experience.
3. We evaluate our design on large-scale NS3 network simulations and show that CLB achieves 17.2% to 42% improvement in reducing the ratio of stalled flows and 15.6% to 47.7% improvement in reducing the ACCT compared with other advanced load balancing schemes and scheduling schemes in three typical data center applications.

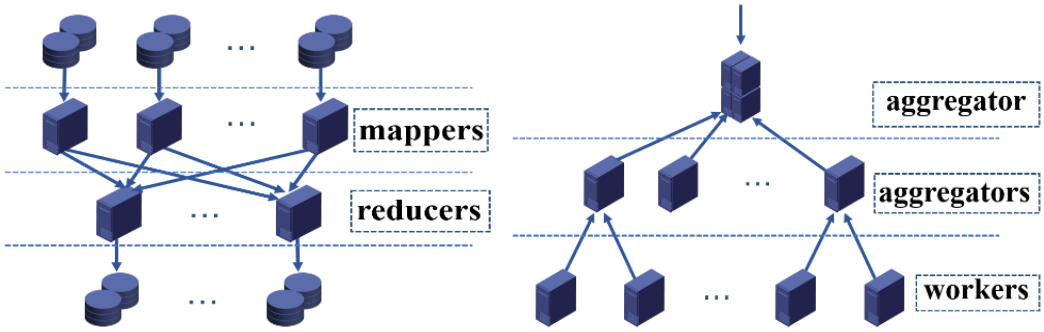
The rest of this paper is organized as follows. We will set out the background and motivation in Section 2. In Section 3, we introduce the details of CLB. The results of large-scale NS3 simulation experiments are presented in Section 4, and Section 5 concludes this paper.

2. Background and Motivation

2.1 Background

2.1.1 The Coflow Abstraction

A Coflow is a collection of flows that shares a common performance goal, such as minimizing the completion time of the latest flow or ensuring that flows meet the common deadline. We assume that the amount of data that needs to be transferred before each coflow starts is known, and the flows of coflow are independent. This is because the input of one flow does not depend on the output of another flow in the same coflow, and the endpoints of these flows can be in one or more machines.



(a) MapReduce **(b)** Aggregator

Fig.1. Examples of communication patterns

Fig.1 shows two typical examples of communication modes in cluster computing applications. Including the shuffle between the mapper and the reducer in Fig.1(a) MapReduce. Additionally, the partition aggregation communication in the online service (such as the user-oriented backend of the social network and search engine) depicted in Fig.1(b). Rather than the transmission times of each flow that makes up the common flow, the task running time is dependent on the transmission time of the total common flow. Because a phase cannot begin or end until all of the data from the phase before it has been received. The flow completion time (FCT) is determined by the transmission time from the start of the task's first flow and the end of its last flow, which makes it apparent that no matter how quickly a single flow is communicated, a task must wait for all of its flows to complete the transmission.

2.1.2 Priority-based Flow Control Mechanism

In PFC, flows are divided into eight priorities according to 802.1Q protocol [16 , 17, 18], as shown in Fig.2. Each priority maintains a queue and implements independent PAUSE mechanism. If the queue length of the input port of the downstream router exceeds the preset threshold XOFF, the downstream

switch will send PAUSE frames to the upstream switch. When the queue length of the ingress port of the downstream switch is lower than another set threshold XON, the downstream switch sends a RESUME frame to the upstream switch to notify it to start sending again. In theory, the lossless DCN can be achieved by reserving enough cache before XOFF to handle packets that arrive before the PAUSE frames takes effect.

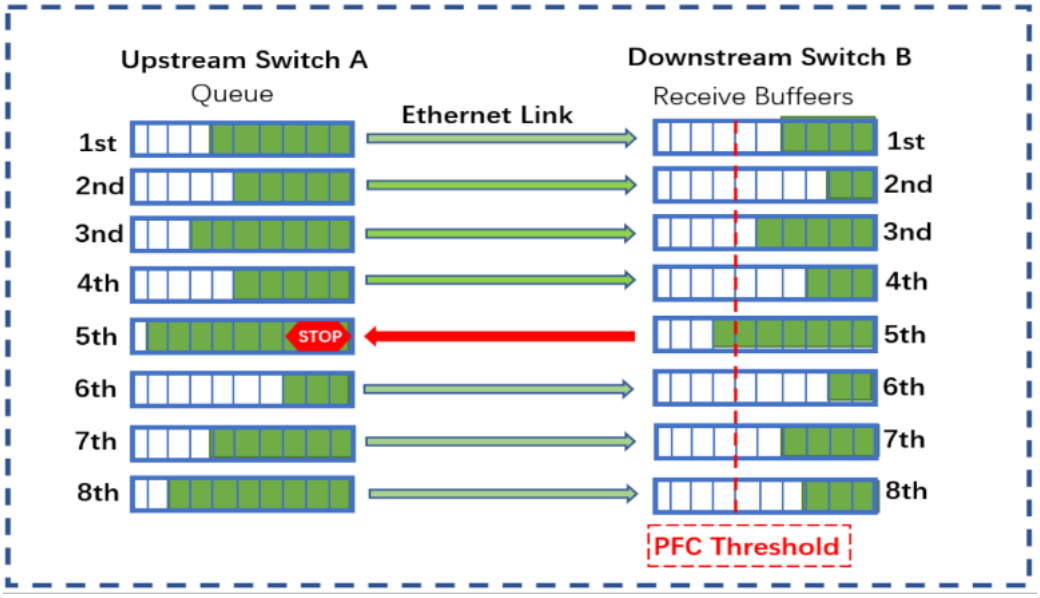


Fig.2. Priority-based Flow Control Mechanism

However, PFC is a coarse-grained mechanism. When a port stops sending, all flows that need to pass through the port will be blocked. This can easily cause the head of the team to block. If a flow in the coflow experience a PFC pausing, it will be regarded as a stalled flow. This behavior will seriously affect the CCT, resulting in serious damage to the communication task performance at this phase.

2.1.3 Existing Load Balancing Schemes in Lossy DCN

A lot of work has been done to better balance the load on lossy DCN [21, 22, 23, 30, 31]. However, for more fine-grained switching schemes, path diversity due to congestion and asymmetry can easily lead to packet reordering. In recent years, many proposals have focused on reducing clutter to improve load balancing. Presto selects a path for a fixed-size stream cell in a polling manner and uses a reordering buffer to reassemble the out-of-order stream cell back in order. CONGA [21] balances flow with flow granularity. If the inactivity gap between flows is greater than the maximum path delay difference, flows can be rerouted without packet reordering. Hermes [23] is resilient to network uncertainty. It only makes a deliberate re-routing decision if it results in a performance improvement. DRILL is a local, per-packet load balancing solution for switches, but congestion mismatches can also occur in the case of topology asymmetry. DRB [24] and RPS [25] are congestion independent load balancing solutions based on each packet/flow unit. In the case of topology asymmetry, they also suffer from congestion mismatch..

Although the above load balancing scheme is effective in lossy DCN, it cannot achieve good performance in lossless DCN due to the PFC mechanism. The reason is that they cannot correctly and timely sense the PFC pausing when selecting the optimal forwarding path. This prompted us to design a new load balancing scheme to avoid stalled flows in coflow.

2.1.4 Coflow Scheduling Schemes

Coflow focuses on a new abstraction that can capture rich task semantics. However, beyond the abstraction, coflow does not propose any new scheduling strategies or mechanisms to achieve Load balance . Based on coflow, Varys adopts the minimum effective bottleneck priority scheduling strategy and minimizes the average coflow completion time by using prior coflow information (i.e., traffic size).

Baraat [12] is a FIFO-based decentralized coflows scheduler that focuses on small coflows. Hedera [26] uses a centralized scheduler to manage traffic to increase network throughput, and MicroTE [27] uses its short-term predictability to adapt to traffic changes. In RAPIER [32], routing and scheduling are considered simultaneously at the coflow level, rather than as separate flows, in order to optimize application performance. Although these coflow scheduling schemes effectively reduce ACCT, they cannot completely avoid the stalling flow phenomenon caused by PFC pausing.

2.2 Motivation

2.2.1 Phenomenon of Traffic Imbalance

Here, we study the characteristics of DCN traffic imbalance and reveal that this characteristic will further deteriorate the PFC mechanism and lead to coflow stagnation. At the same time, we specifically analyze the characteristics of transmission imbalance in space, time and scale.

1) Spatial Imbalance

In the process of traffic transmission in the existing DCN, the traffic size and congestion through different edge switches are often different. Therefore, if the coflow of the same task passes through different switches, the congestion and queuing delay will be different. This may lead to a very congested path in the transmission process, or even PFC pausing.

2) Temporal Imbalance

With the wide application of virtualization in the DCN, processor sharing and virtual machine coexistence will aggravate the time imbalance in the DCN [33, 34, 35, 36]. In addition, the terminal host delay in the DCN usually follows the Poisson distribution between 0 ms and 100 ms, which will further lead to path congestion. In this way, the ratio of stalled flows will continue to worsen.

3) Size Imbalance

Different DCN have different traffic scales. In the flow size distribution in the common DCN, nearly 80% of the traffic size is less than 80KB, and 20% of the traffic size is greater than 80KB. Obviously, assuming that the number of concurrent flows is increased on the traffic distribution of this scale proportion, the probability of the links being suspended by PFC will be greatly increased, resulting in the problem of stalled flows becoming more serious.

2.2.2 Why PFC Leads to the Stalled Flows Problem?

Due to the existing load balancing schemes cannot correctly and timely sense PFC pausing, the performance in the lossless DCN is poor. In order to better explain the influence of the PFC pausing leading to the stalled flows in coflows, we specially design an example and demonstrate how PFC mechanism leads to stalled flows phenomenon.

Without loss of generality, as shown in Fig.3, the topology scenario we set is the common leaf-spine topology in the DCN. The senders (H1, H2, H3) and receivers (R1, R2, R3) are connected to the corresponding edge layer leaf switches (L1, L2), respectively. There are three equal-cost paths between two leaf switches, each link bandwidth is 40Gbps and link delay is 5μs, the switch buffer is set to 9MB. The PFC pausing threshold of the ingress port and egress port of each switch is set to 256KB. Meanwhile, we use the congestion control protocol DCQCN, where the parameters are set to the recommended default values. In the end, we use NS-3 network simulation platform to show the appearance of the stalled flows phenomenon in coflows caused by PFC pausing.

In this test, the coflow contains two flows, flow A1 and flow A2 respectively, which are sent from senders H1 and H2 to receivers R1 and R2 respectively. Meanwhile, sender H3 sends a congested flow B to receiver R3, and flow A1 is transmitted through the path from L1 to S1. flow B transmits through the path from L1 to S2, and the transmission of the congested flow is about to trigger the PFC pausing threshold. If flow A2 selects the path from L1 to S2 for transmission, the L2 switch will reach the PFC pausing threshold and send PAUSE frames to S2. As a result, the path from L1 to S2 is paused and flow A2 becomes a stalled flow, even though flow A1 has already reached R1. Since flow A2 becomes a stalled flow, tasks must wait for flow A2 to arrive before they can complete or proceed to the next task. Because of flow A2 fails to sense that the link from L1 to S2 is likely to experience PFC pausing, flow A2 becomes a stalled flow, as shown in Fig.4(a) and (b). The two figures reveal the impact of traffic experience PFC pausing on CCT. The PFC pausing resulted in a significant increase in the CCT.

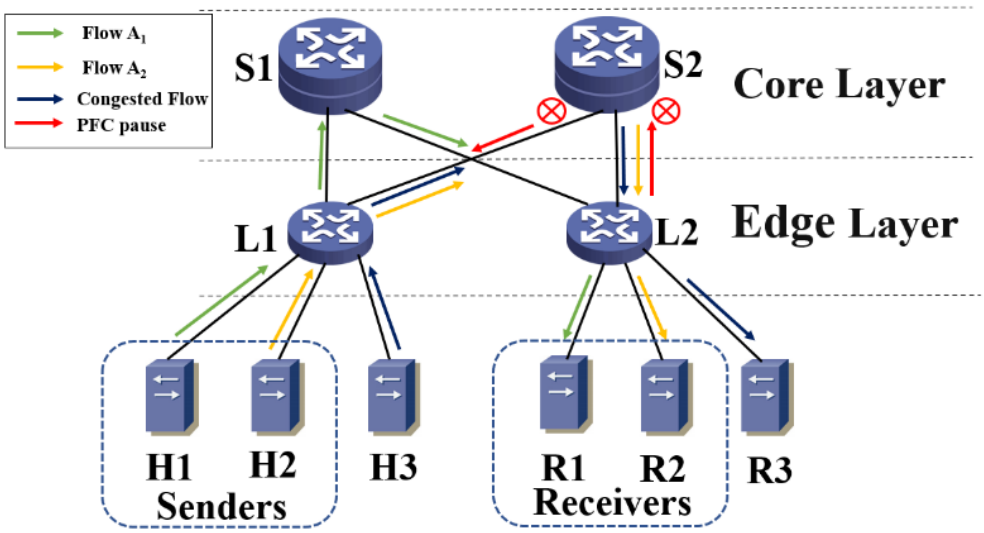
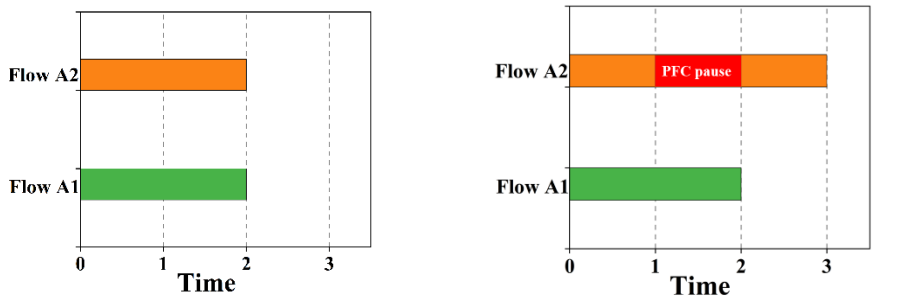


Fig.3. In the typical leaf-spine structure topology of DCN, coflow (FA1, FA2) is sent from the source host (H1, H2) to the destination host (R1, R2) through two different paths, while FA2 becomes a stalled flow because it selects the path that experience PFC pausing.



(a) FA2 no experience PFC pausing(CCT=2) **(b)** FA2 becomes stilled flow(CCT=3)

Fig.4. The impact of the PFC mechanism

2.2.3 Summary

In this section, through theoretical analysis and demonstration of examples, we draw the following conclusions:

1) The traffic characteristics of the DCN include time imbalance, space imbalance and scale imbalance. Therefore, in the lossless DCN scenario, it is likely to cause PFC pausing of the links.

2) In the lossless DCN environment, the failure to perceive whether the links may have PFC pausing is the reason for the failure of the existing classical load balancing and coflow scheduling mechanisms.

These conclusions lead us to present a new load balancing scheme that combines the flows characteristics of coflows and considers the impact of PFC pausing on coflows. In the remainder of this paper, we describe our design and its implementation.

3. The CLB Design

3.1 Design Rationale

3.1.1 Basic Idea

First, we introduce the core idea of CLB. As described above, the existing load balancing and coflow scheduling mechanisms cannot sense the possible PFC pausing on the links in the lossless DCN. And their rerouting decision cannot solve the problems of stalled flows in coflows caused by PFC pausing in the lossless DCN. PFC pausing will inevitably lead to serious packet reordering and large queue delay, and indirectly lead to the phenomenon of stalled flows. Therefore, in order to solve this problem, we provide a new load balancing scheme that can effectively reduce the ratio of stalled flows in the traffic transmission process. CLB combines the traffic characteristics of coflows to support the high performance of coflows in the lossless DCN.

In order to ensure the efficient communication transmission of coflows among multiple server clusters, and consider simultaneously the ease of deployment and implementation of the load balancing scheme, we deploy the CLB load balancing scheme on switchs. Through Route forwarding module, we avoid flows select the path with high probability experencing PFC pausing, so as to avoid the occurrence of stalled flows in coflows. We obtain each egress queues size and the change rate of egress queues size on switch. The two reference indicators are strongly related to whether PFC pausing is triggered on the links. And the indicators information can be directly obtained from switch.

3.1.2 Design Overview

The overview of CLB is shown in Fig.5, and the followings are a brief introduction to the route classification module and route forwarding module.

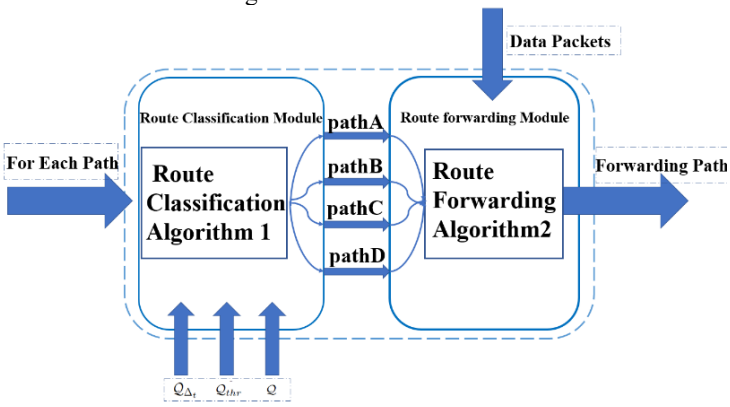


Fig.5. CLB overview

1) Route classification module

In route classification process, there involve three parameters, include the egress queue size Q, the preset threshold that lower than PFC pausing threshold Q_{thr} ,and the derivative of the egress queue size $Q_{\Delta t}$. These three parameters are strongly related to whether links experienced PFC pausing. Each port on the switch is classified by route classification algorithm 1, and the path is divided into four types of paths, so that the load balancing decision can be made on route forwarding module.

2) Route forwarding module

In route forwarding process, based on the results obtained by the route classification module, the traffic makes a load balancing decision in this module through route forwarding module algorithm 2. It will preferentially select the high priority port determined in the route classification module for routing and forwarding, and comprehensively consider the traffic characteristics of coflows, which will greatly reduce the probability of traffic expereenced PFC pausing during communication transmission.

3.2 Design Details

3.2.1 Route Classification Details

As shown in Fig.6, we maintain four tables on each switch, the table T1 record switch the queue size of egress ports $\mathcal{Q}$, the switch can obtain the parameter $\mathcal{Q}$ directly, the table T2 record the derivative of the egress queue size $\mathcal{Q}_{\Delta t}$, the switches calculate $\mathcal{Q}_{\Delta t}$ in a certain interval (small) changes of the egress queue size, The table T3 records the traffic distribution of coflows in each port $\mathcal{T}r$, and the table T4 record the route classification results. Based on these three parameters, switchs classify each reachable port into four types for load balancing decisions.

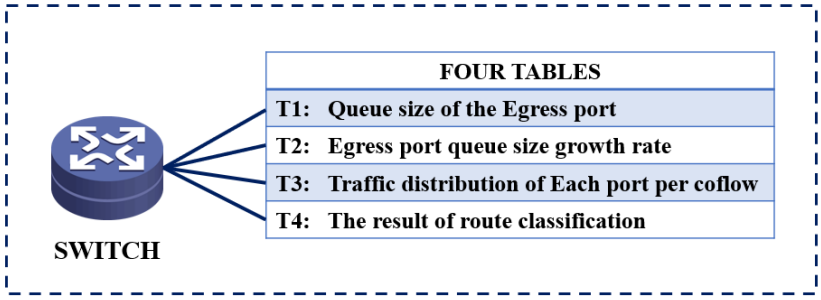


Fig.6. Four tables are maintained on the switch

Algorithm 1 Route Classification

Input: $\mathcal{Q}_{\Delta t}$: The derivative of the queue size;
$\mathcal{Q}_{thr}$: The preset threshold that lower than the PFC pausing threshold;
$\mathcal{Q}$: Indicates the current queue size of the port;
type : Characterization of path condition;
Output: Priority of each port;
1: **for** each port and priority $\mathcal{p}$ **do**
2: **if** $\mathcal{p}.\mathcal{Q} < \mathcal{Q}_{thr}$ and $\mathcal{p}.\mathcal{Q}_{\Delta t} \leq 0$ **then**
3: *type* = pathA;
4: **else if** $\mathcal{p}.\mathcal{Q} < \mathcal{Q}_{thr}$ and $\mathcal{p}.\mathcal{Q}_{\Delta t} > 0$ **then**
5: *type* = pathB;
6: **else if** $\mathcal{p}.\mathcal{Q} \geq \mathcal{Q}_{thr}$ and $\mathcal{p}.\mathcal{Q}_{\Delta t} \leq 0$ **then**
7: *type* = pathC;
8: **else if** $\mathcal{p}.\mathcal{Q} > \mathcal{Q}_{thr}$ and $\mathcal{p}.\mathcal{Q}_{\Delta t} > 0$ **then**
9: *type* = pathD;
10: **return** the *type* of each path;

As shown in algorithm 1, switchs classify each reachable port into four types. The first type of port is PathA: the egress port queue size $\mathcal{p}.\mathcal{Q}$ does not exceed the preset egress port queue threshold $\mathcal{Q}_{thr}$, and the queue size growth rate $\mathcal{p}.\mathcal{Q}_{\Delta t} \leq 0$. This type of port is the best route forwarding port, with the minimum probability of experencing PFC pausing, and belongs to the good path.

The second type of port is PathB: the egress port queue size $\mathcal{p}.\mathcal{Q}$ does not exceed the preset egress port queue threshold $\mathcal{Q}_{thr}$, and the queue growth rate $\mathcal{p}.\mathcal{Q}_{\Delta t} > 0$. According to the growth rate, it can be roughly judged that this type of port may trigger PFC pausing in the future, leading to the appearance of stalled flows. This type of port is the suboptimal path,

The third type of port is PathC: the egerss queue size $\mathcal{p}.\mathcal{Q}$ has exceed the preset egerss queue threshold $\mathcal{Q}_{thr}$ but not reach the PFC pausing threshold, and queue growth rate $\mathcal{p}.\mathcal{Q}_{\Delta t} \leq 0$, this kind of path compared with the second path though queue growth rate $\mathcal{p}.\mathcal{Q}_{\Delta t}$ is negative, but had reached the preset the egress queue threshold $\mathcal{Q}_{thr}$. Result in are more likely to experience PFC pausing comparing with PathB.

The fourth and last type of path is PathD: the egress port queue size $p.Q$ has reached the preset egress port queue threshold Q_{thr}, and the queue growth rate $p.Q_{\Delta t} > 0$. This type of path has the highest probability of encountering PFC pausing, PathD is the congestion path among these four types of paths, even the worst path.

3.2.2 Route Forwarding Details

In order to prevent traffic from encountering PFC pausing as much as possible, we make load balancing decisions based on the route classification results recorded in table T4, and preferentially select port with low probability of experencing PFC pausing to avoid traffic from becoming stalled flows.

As shown in algorithm 2, among all the paths that can reach the receivers, we tend to preferentially choose PathA, which is the optimal path and has the smallest probability of experencing PFC pausing. In the same PathA type of candidate port, switchs preferentially chooses the port with the smallest queue length. If there is no PathA, we preferentially choose the PathB with the smallest queue growth rate among the PathB for routing and forwarding. Similarly, If there is no PathB, we choose the PathC path with the smallest difference between the queue length size and the preset threshold in the PathC. In the worst case, if there is only PathD, we choose the PathD with the smallest queue growth rate in the PathD for routing and forwarding.

Algorithm 2 Route Forwarding

for every packet **do**
Assume its corresponding path is p
$\{p'\}$ = all pathA
$\{p''\}$ = all pathB
$\{p^3\}$ = all pathC
$\{p^4\}$ = all pathD
$\{p^c\}$ = all candidate path
    if $\{p'\} \neq \emptyset$ **then**
        $\{p^c\} = Min_{p \in \{p'\}}(p.Q)$;
    else if $\{p''\} \neq \emptyset$ **then**
        $\{p^c\} = Min_{p \in \{p''\}}(p.Q_{\Delta t})$;
    else if $\{p^3\} \neq \emptyset$ **then**
        $\{p^c\} = Min_{p \in \{p3\}}(p.Q)$;
    else if $\{p^3\} \neq \emptyset$ **then**
        $\{p^c\} = Min_{p \in \{p4\}}(p.Q_{\Delta t})$;
    if *card* $\{p^c\} \geq 2$ **then**
        $p^* = Min_{p \in \{pc\}}(p.T_r)$
return p^*

If there are candidate ports with the same priority, according to table T3, switchs can obtain the traffic distribution of all coflows at each port ($p.T_r$), we preferentially select the port that is not selected or less flows passes through in the coflows during the transmission process of coflows, so as to avoid the phenomenon that if a path experiences PFC pausing, most flows in one coflow will become stalled flows at the same time, leading to more traffic congestion. Switchs distribute the internal coflows traffic on each port in a balanced way to improve the link utilization and prevent most traffic from becoming stalled flows, result in achieving high performance.

3.2.3 Parameter Setting

The preset queue size threshold Q_{thr} , theoretically, cannot be set too large or too small. Otherwise, the route classification will fail to make the port distinguished and the effect of route classification will be weakened.

If the setting is too large, it will lead to the partition of ports are all pathA and pathB ports, all paths are good paths, but it is obvious that this should not be the case in reality. If the threshold is too small, all the links divided will be pathC and pathD links, and all the paths are bad paths, which is also not in line with the reality.

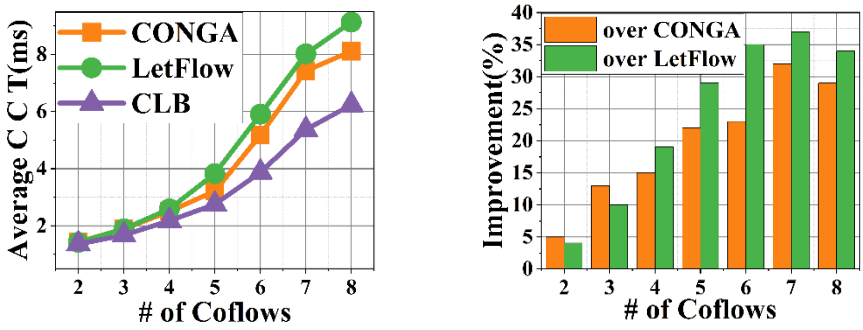
So we should choose a suitable default port queue size threshold, all ports can have distinct administrative levels feeling and the degree of differentiation, through massive experiments, we suggest that the threshold for the default value of two-thirds of the PFC pausing threshold, if PFC pausing threshold is 256 KB, then this should be the default threshold for 170 KB, It can make the load balancing scheme have better effect.

4. Evaluation and Analysis

On the NS-3 network simulation platform, we compare CLB with three typical load balancing schemes (Hermes, CONGA, LetFlow) and a typical traffic scheduling scheme Varys. And we conduct small-scale basic performance test experiments and large-scale application performance test experiments on CLB.

4.1 Baseline

Simulation Settings: The topology we used in the baseline basic performance test experiment is the motivation example topology (Fig.3), which has two leaf switches and two spine switches. And each leaf switch connects three hosts, the links bandwidth is 40Gbps, the links delay is set to 5 μs, and the over-subscription rate of the leaf layer is 3:2. Besides, we use DCQCN [16] as the default transport protocol and set the relevant parameters in the proposal. Meanwhile, all switches enable PFC mechanism and has a shared buffer of 6MB.The senders send n coflows (increasing by load strength) to receivers at the same time. Each coflow is composed of n/2 concurrent flows, and the size of each flow is evenly distributed between 10KB and 100KB.



(a) Coflow's completion time **(b)** Improvement of CLB

Fig.7. Performance testing in small-scale topologies

By comparing with the classical load balancing protocols CONGA [21] and LetFlow [22], we investigated the excellent performance of CLB in the motivation example topology. Fig.7(a) and (b) show the influence of CLB on ACCT under different loads. The results show that with the increase of coflows, CLB significantly improves the network performance. CLB can significantly reduce the ACCT of coflows in the lossless DCN environment with PFC mechanism. Compared with CONGA and LetFlow load balancing mechanism, the performance improvement of CLB can be up to 32% and 37% under different load strengths. This is because compared with CONGA and LetFlow load balancing mechanisms, CLB can timely sense the possible PFC pausing of the links and reduce the frequency of PFC pausing triggering. It also makes the load balancing decision that minimizes the probability of coflows experiences PFC pausing to avoid stalled flows on links in time. CLB allocates flows in each coflow reasonably to prevent

most flows from becoming stalled flows at the same time, effectively reduces out-of-order packets, and ACCT.

4.2 Large-Scale Application Performances

In this section, we conduct a large number of experiments through the NS-3 network simulation platform. Effectively verify the performance of CLB in DCN applications in large-scale scenario. Such as Map-Reduce application, Cloud storage application, Web search application [37, 38], etc. Compared with several classical load balancing schemes, including Hermes [23], CONGA [21], and the classical coflow scheduling scheme Varys [11], the experimental results show that CLB has better performance in reducing ACCT and the ratio of stalled flows.

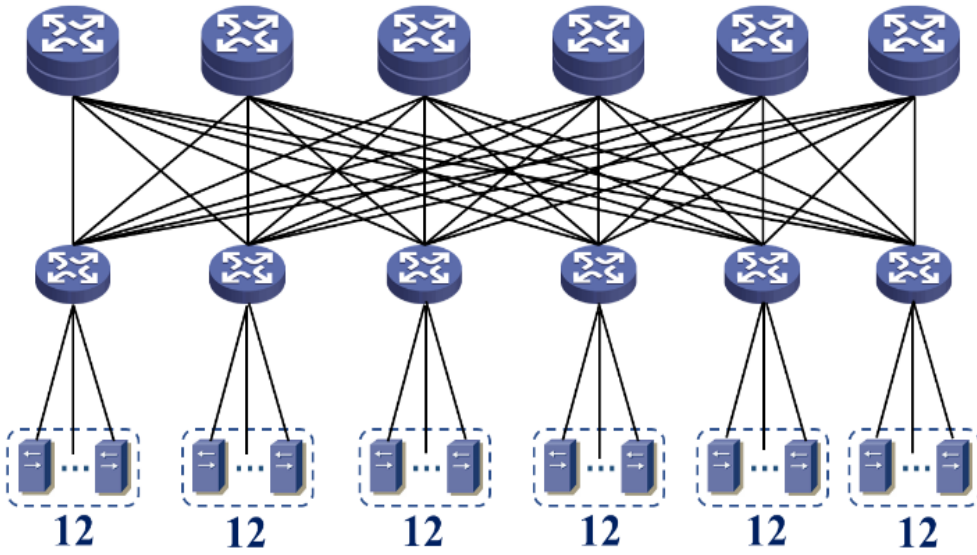


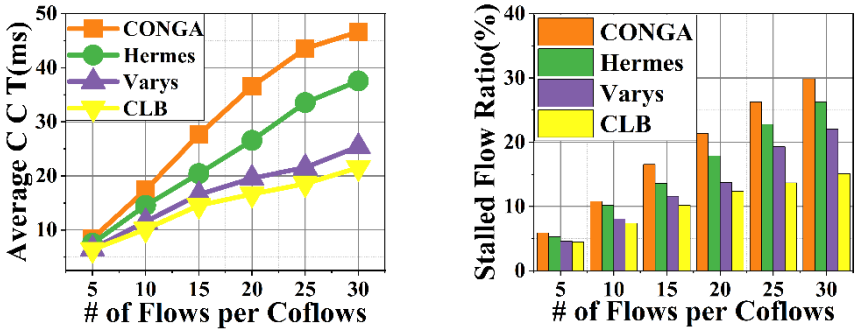
Fig.8. Large-Scale leaf-spine network topology

Unless otherwise stated, the following experiments adopt the classic leaf-spine topology of the lossless DCN. As shown in Fig.8, in this topology, the number of spine switches is 6, and the number of leaf switches is 6. Meanwhile, each leaf switch is connected with 12 hosts, and there are 6 equivalent parallel paths between any pair of leaf switches. Moreover, all links bandwidths are set to 40Gbps and the links delay is set to 5 μs. The PFC mechanism [17] is enabled on all switches, and the shared buffer area is 6MB. We use DCQCN as the default transmission protocol and set relevant parameters as recommended.

4.2.1 Map-Reduce Application

First, we simulate the performance of Hermes, CONGA, Varys and CLB in Map-Reduce applications. We set 30 coflows with the same amount of flows, each flow with the same size of 16KB, and the flow arrival time following a Poisson distribution with parameter λ=2.5. We measured the ACCT and the proportion of stalled flows under different amounts of flows in coflows by gradually increasing the number of flows in each coflow from 5 to 30.

It can be seen from Fig.9(a) that in Map-Reduce application scenario, compared with CONGA, Hermes and Varys, the ACCT of CLB is reduced by 39%, 30.2% and 15.6% respectively under different load strengths. Similarly, the results in Fig.9(b) show that the stalled flow ratio can be reduced by 34.4%, 25.3% and 14.3% respectively.



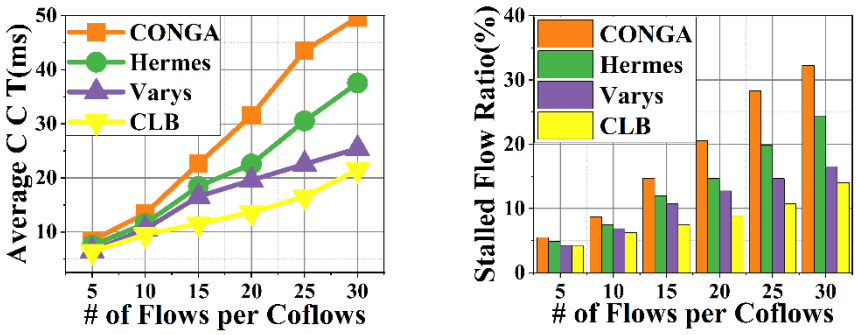
(a) Coflow's Completion Time **(b)** Stalled Flows Ratio

Fig.9. Map-Reduce Application

4.2.2 Cloud Storage Application

In the scenario of Cloud storage applications, we fix that each coflows only contains 15 flows. All flows are randomly distributed among the senders, the arrival time of the flows following a Poisson distribution with parameter λ =2.5. Meanwhile, the flow size distribution following the Pareto distribution, and the size is limited between 10KB and 2MB. Through increasing the number of tasks in the network from 5 to 30, we measured the impact of CLB on the perceived traffic load, and analyzed the experimental results with other protocols.

It can be seen from Fig.10(a) that in Cloud Storage Application scenario, compared with CONGA, Hermes and Varys, the ACCT of CLB is reduced by 46.2%, 33.1% and 19.1% respectively under different load levels. At the same time, the results in Fig.10(b) show that the stalled flow ratio can be reduced by 41.5%, 31.5% and 17.2% respectively.



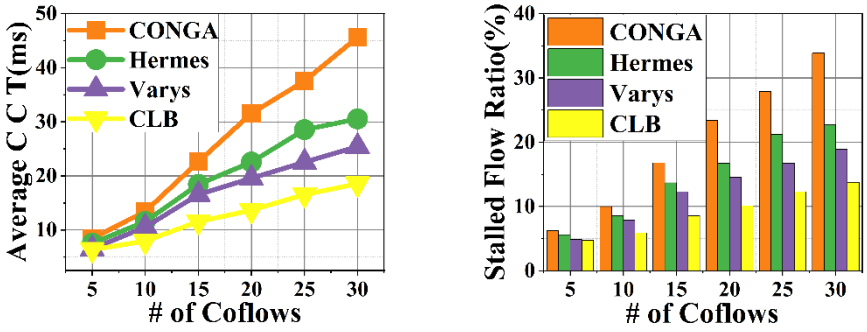
(a) Coflow's Completion Time **(b)** Stalled Flows Ratio

Fig.10. Cloud Storage Application

4.2.3 Web Search Application

In the Web search application scenario [37, 38], we set 30 concurrent coflows, and the flows size is set to 2MB/n, and n is the number of flows in each coflow. The arrival time of all traffic are randomly distributed, and flow is also randomly distributed among senders. Similarly, we also measure the ACCT and stalled flow ratio of several different load balancing schemes by gradually increasing the number of flows n in each coflow from 5 to 30.

It can be seen from Fig.11(a) that in Web search application scenario, compared with CONGA, Hermes and Varys, the ACCT of CLB is reduced by 47.7%, 34.1% and 23.6% respectively under different load levels. At the same time, the results in Fig.11(b) show that the stalled flows ratio can be reduced by 42%, 29.4% and 20.3% respectively.



(a) Coflow's Completion Time **(b)** Stalled Flows Ratio

Fig.11. Web Search Application

Analysis: The research results show that compared with other load balancing protocols, CLB can significantly improve the transmission efficiency of coflows links. Moreover, the CLB protocol can effectively prevent flows from becoming stalled flows due to PFC pausing on the links, thus achieving the effect of reducing ACCT. Through large-scale experiments in this section, CLB has excellent performance in the lossless DCN with PFC mechanism. The most critical reason is that CONGA makes load balancing decisions based on link utilization. However, PFC pausing of the links will also lead to low link utilization, thus misleading CONGA to take wrong load balancing decisions. It even affects more flows to become stalled flows, and the overall performance of the DCNs is greatly reduced. Although Hermes uses ECN and RTT information to make load balancing decisions, it still cannot effectively perceive that the links may experience PFC pausing, which inevitably leads to the generation of stalled flows. Although Varys makes flows scheduling decisions based on coflows, its performance is better than Hermes and CONGA. At the same time, it does not consider that the links may experience PFC pausing, nor can it avoid causing the flows to become stalled flows. On the contrary, CLB has a strong sense of the links may experience PFC pausing, and has a strong dynamic adjustment performance. At the same time, CLB also consider the traffic characteristics of coflows, and flows of the same coflows are evenly distributed to the links. Even if a link experience PFC pausing, it is possible to avoid the phenomenon that most of the flows becomes stalled flows at the same time. In general, the CLB mechanism significantly improves the transmission efficiency of coflows in the lossless DCN.

5. Conclusion

In this paper, we design and implement CLB, a lightweight and user-centered load balancing scheme. It combines the traffic characteristics of coflow and avoids the traffic of coflow from becoming stagnant in the lossless DCN through the immediate perception of PFC pausing on the links. CLB can be flexibly deployed in the existing lossless DCN, and successfully avoids the complex implementation and deployment of other load balancing schemes. The experimental results show that CLB can effectively reduce the proportion of stalled flow and significantly reduce the ACCT by 30-50% compared with other schemes in the typical application of DCN, significantly in reducing response latency and improve the user experience.

Author's Contributions

Jin Wang contributed significantly to analysis and manuscript preparation; Yi He performed the experiment and the data analyses and wrote the manuscript; Jinbin Hu helped perform the analysis with constructive discussions; Shuying Rao performed the algorithm design; Wangqing Luo contributed to the conception of the study.

Funding

This work is supported by the National Natural Science Foundation of China (62102046), and the Natural Science Foundation of Hunan Province (2022JJ30618).

Competing Interests

The authors declare that they have no competing interests.

References

[1] Foster, Y . Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” in 2008 Grid Computing Environments Workshop. IEEE, 2008, pp. 1–10.

[2] G. Liu and H. Shen, “An Economical and SLO-Guaranteed Cloud Storage Service across Multiplee Cloud Service Providers,” in INFOCOM’16, 2016.

[3] Jin Wang, Liu Wang, Shiming He, Osama Alfarraj, Amr Tolba, R. Simon Sherratt, SA-RFR: Self-attention based Recurrent Feature Reasoning for Image Inpainting with Large-missing Area, Human-centric Computing and Information Sciences.

[4] Y . Y u and C. Qian, “Space shuffle: A scalable, flexible, and high-bandwidth data center network,” in 2014 IEEE 22nd In- ternational Conference on Network Protocols. IEEE, 2014, pp. 13–24.

[5] V . S. Rajanna, A. Jahagirdar, S. Shah, and et al, “Explicit coordination to prevent congestion in data center networks,” Cluster Computing, vol. 15, pp. 183–200, June 2012.

[6] M. Chowdhury, Z. Liu, A. Ghodsi, and I. Stoica, “HUG: Multi-resource fairness for correlated and elastic demands,” in 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), 2016, pp. 407–424.

[7] U. Hoelzle, J. Dean, and L. A. Barroso, “Web Search for A Planet: The Arichitecture of the Google Cluster,” IEEE Micro Magazine, April 2003.

[8] B. V amanan, J. Hasan, and T. N. Vijaykumar, “Deadline-Aware Datacenter TCP (D2TCP),” in SIGCOMM’12, 2012.

[9] Jin Wang, Hui Han, Hao Li, Shiming He, Pradip Kumar Sharma, Lydia Chen, Multiple Strategies Differential Privacy on Sparse Tensor Factorization for Network Traffic Analysis in 5G, IEEE Transactions on Industrial Informatics, vol.18, no.3, pp.1939-1948, 2022.

[10] M. Chowdhury et al. Coflow: A networking abstraction for cluster applications. In HotNets-XI, pages 31–36. 2012.

[11] M. Chowdhury, Y . Zhong, and I. Stoica. Efficient coflow scheduling with Varys. In SIGCOMM, 2014.

[12] F. Dogar, T. Karagiannis, H. Ballani, and A. Rowstron. Decentralized task-aware scheduling for data center networks. In SIGCOMM, 2014.

[13] M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica. Managing data transfers in computer clusters with Orchestra. In SIGCOMM, 2011.

[14] G. Ananthanarayanan, S. Kandula, A. Greenberg, and et al, “Reining in the Outliers in Map-Reduce Clusters using Mantri,” in OSDI’10, 2010.

[15] Jin Wang, Caiyan Jin, Naixue Xiong, Qiang Tang, Gautam Srivastava, Intelligent Ubiquitous Network Accessibility for Wireless-Powered MEC in UAV-Assisted B5G, IEEE Transactions on Network Science and Engineering, vol.8, no.4, pp.2801-2813, 2021.

[16] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang. Congestion Control for Large- Scale RDMA Deployments. In Proc. ACM SIGCOMM, 2015.

[17] IEEE 802.1 Qbb - Priority-based Flow Control. https://1.ieee802.org/ dcb/802-1qbb/.

[18] C. Tian, B. Li, L. Qin, J. Zheng, J. Yang, W. Wang, G. Chen, and W. Dou. P-PFC: Reducing Tail Latency with Predictive PFC in Lossless Data Center Networks. IEEE Transactions on Parallel and Distributed Systems, 31(6):1447-1459, 2020.

[19] Y. Lu, G. Chen, B. Li, K. Tan, Y. Xiong, P. Cheng, J. Zhang, E. Chen, and Thomas Moscibroda. multipath Transport for RDMA in Datacenters. In Proc. USENIX NSDI, 2018.

[20] K. Chudgar and S. Sathe. 2014. Packet Forwarding System and Method Using Patricia Trie Configured Hardware. (2014). http://www.google.com/patents/ US8767757 US Patent 8,767,757.

[21] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, G. Varghese. CONGA: Distributed Congestion-Aware Load Balancing for Datacenters. In Proc. ACM SIGCOMM, 2014.

[22] E. Vanini, R. Pan, M. Alizadeh, P. Taheri and T. Edsall. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. In Proc. USENIX NSDI, 2017.

[23] H. Zhang, J. Zhang, W. Bai, K. Chen, and M. Chowdhury. Resilient Datacenter Load Balancing in the Wild. In Proc. ACM SIGCOMM, 2017.

[24] Jiaxin Cao, Rui Xia, Pengkun Yang, Chuanxiong Guo, Guohan Lu, Lihua Yuan, Yixin Zheng, Haitao Wu, Yongqiang Xiong, and Dave Maltz. Per-packet Load-balanced, Low-latency Routing for Clos-based Data Center Networks. In CoNEXT 2013.

[25] Advait Dixit, Pawan Prakash, Y Charlie Hu, and Ramana Rao Kom- pella. On the Impact of Packet Spraying in Data Center Networks. In INFOCOM 2013.

[26] M. Al-Fares et al. Hedera: Dynamic flow scheduling for data center networks. In NSDI. 2010.

[27] T. Benson et al. MicroTE: Fine grained traffic engineering for data centers. In CoNEXT. 2011.

[28] K. Ramakrishnan, S. Floyd, and D. Black, “The addition of explicit congestion notification (ecn) to ip,” Tech. Rep., 2001.

[29] N. Dukkipati, T. Refice, Y. Cheng, and et al, “An argument for increasing TCP’s initial congestion window,” Computer Communication Review, vol. 40, no. 3, pp. 26–33, 2010.

[30] J. Hu, J. Huang, W. Lv, Y. Zhou, J. Wang and T. He. CAPS: Coding- based Adaptive Packet Spraying to Reduce Flow Completion Time in Data Center. IEEE/ ACM Transactions on Networking, 2019, 27(6): 2338-2353.

[31] J. Hu, J. Huang, Z. Li, Y. Li, W. Jiang, K. Chen, J. Wang and T. He. RPO: Receiver-driven Transport Protocol Using Opportunistic Transmission in Data Center. In Proc. IEEE ICNP, 2021.
[32] Y. Zhao, K. Chen, W. Bai, C. Tian, Y. Geng, Y. Zhang, D. Li, and S. Wang. RAPIER: Integrating routing and scheduling for coflow-aware data center networks. In INFOCOM, 2015.
[33] L. Suresh, M. Canini, S. Schmid, and A. Feldmann, "C3: Cutting Tail Latency in Cloud Data Stores via Adaptive Replica Selection," in NSDI'15, 2015.
[34] X. Zhang, E. Tune, R. Hagmann, R. Jnagal, V . Gokhale, and J. Wilkes, "CPI2: CPU performance isolation for shared compute clusters," in EuroSys, 2013.
[35] M. Kambadur, T. Moseley, R. Hank, and M. Kim, "Measuring Interference Between Live Datacenter Applications," in SC, 2012.
[36] G.Wang and T. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," in IN-FOCOM'10, San Diego, CA, USA, March 2010.
[37] W. Bai, L. Chen, K. Chen, D. Han, C. Tian, and H. Wang. Information-Agnostic Flow Scheduling for Commodity Data Centers. In Proc. USENIX NSDI, 2015.
[38] S. Hu, W. Bai, G. Zeng, Z. Wang, B. Qiao, K. Chen, K. Tan, and Y. Wang.Aeolus: A Building Block for Proactive Transport in Datacenters. In Proc. ACM SIGCOMM, 2020.