Journal of Network Computing and Applications (2016) 1: 12-20
Clausius Scientific Press, Canada

Flow Scheduling Strategies for Minimizing Flow Completion Times in Information-agnostic Data Center Networks

Peiyi Yu[1,a], Chao Hu[12*,b], Bo Liu[1], Changyou Xing[1]

[1]College of Command Information Systems, PLA University of Science and Technology, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education
[a]ypy02784@163.com, [b] huchaonj@126.com

Keywords: completion time; scheduling strategy; data center networks; information agnostic

Abstract: Minimizing the flow completion time (FCT) is widely considered as an important optimization goal in designing data center networks. However, existing schemes either rely on the precondition that the size and deadline of each flow is known in advance, or require modifying the switch hardware, which is hard to implement in practice. In this paper, we present MCPF, a flexible and dynamic flow scheduling strategy to reduce the FCT. This strategy is based on the estimated probabilities of each flow to finish the transmission in a period time, and these flows which have higher completion probabilities are assigned with higher priority. Meanwhile, switches perform flow scheduling according to these priorities. We employ a queueing theory based mathematical model to analyze the average FCT of MCPF, and compare it with other two flow scheduling strategies. We also introduce the challenges and the solutions to implement MCPF in realistic networks. Finally, we evaluate the performance of MCPF in Mininet. The analysis and experimental results show that MCPF could effectively reduce the FCT.

1. Introduction

Online services, e.g. web search, social networks, retail, have become an indispensable part of human society. These applications supply users with practical and convenient services, which promote the work and life for people. As the infrastructure of online services, data center networks are in an important position, and take charge of information processing and query response. Due to the interactive characteristic, lots of online services have rigorous demands on response time, and the deadlines are usually less than several hundred milliseconds. For example, the 99.9% of responses in web site are required to complete in 200-300 milliseconds [1]. From the perspective of traffic pattern in data center networks, most of flows are short, but the elephant flows generated by replication and virtual machine migration etc. occupy high proportion of bytes [2]. Comparing with elephant flows, mice flows are generally from interactive foreground services, and they are sensitive to the network latency. These flows should be finished in a short time, while elephant flows don't need to satisfy the deadline. When an elephant flow is in a link, the transmission of short flows will be seriously affected [4]. Therefore, a prime purpose of designing data center

networks is to reduce the flow completion time (FCT) of short flows, and improve the throughput of elephant flows.

In this paper, we introduce a novel flow scheduling strategy, namely MCPF (Maximal Completion Probability First), and it employs "multi-queue & flow completion probability based priority". In MCPF, we also make use of multiple queues in commodity switches like PIAS [6], but the priorities of each flow are determined by the probabilities of flows being completed transmission in a period time rather than flow size, and the flows having higher completion probabilities are granted higher priorities, so they can be accelerated when they are close to finish transmission. During packet forwarding, switches firstly serve these packets having the highest priority. Some short flows may be a little affected, but other flows can shorten their queueing delay.

It's worthy of noting that MCPF strategy is an approach that can decrease FCT through flexible and dynamic flow scheduling in switches, and it is not incompatible with other types of schemes, such as DCTCP [3] and MPTCP [10]. In fact, it can further improve the performance of data center networks by synthesizing with other schemes.

2. MCPF Overview

MCPF is also an information-agnostic dynamic flow scheduling strategy like PIAS, and it uses "multi-queue & flow completion probability based priority". The primary conception of MCPF is flow completion probability, which reflects the possibility that a flow finish its transmission in a period time, and the order of flow scheduling depends on these probabilities of each flow. To simplify the description of MCPF strategy, we define the following notations in Table 1.

Table 1. Notations

Notations	Definitions
f(x)	The probability density distribution function of flow size in data center network, where x is the number of packets and $x \in [1,+\infty)$
t	The time interval to perform flow scheduling in switches
N	The number of flows in switches when switches start to schedule flows
Bi	The bytes that flow i has sent
C	The capacity of switches
pi	The probability of flow i that completes transmission in the next time interval

According to the above definitions and conditional probability formula, the probability of flow i can finish transmitting in t is

$$\begin{aligned} p_i &= P(flow\ size \le B_i + Ct \mid flow\ size > B_i) \\ &= \frac{P(B_i < flow\ size \le B_i + Ct)}{P(flow\ size > B_i)} \\ &= \frac{\int_{B_i}^{B_i+Ct} f(x)dx}{\int_{B_i}^{+\infty} f(x)dx} \end{aligned} \tag{1}$$

Notice that flow size distribution is discrete, we simplify it with fluid model. Based on Eq. 1, we can obtain the completion probabilities of each flow, and the flow whose probability is the highest is firstly served. If this flow finish transmission before the next flow scheduling, the flow with second highest probability are scheduled. In addition, we set the priority of a new flow to be highest

so that the short flows can be ensured to finish transmission as soon as possible. The detail of MCPF strategy is showed in Fig. 1.

Algorithm: Maximum Completion Probability First algorithm
Input: B_i, t
Output: The order of flow scheduling

1. **for** $1 \leqslant i \leqslant N$
2. calculate all flows' probabilities of flow completion in t through Eq. (1);
3. **end for**
4. sort all flows according to their probabilities of flow completion in descending order;
5. **while** the scheduling cycle is not over
6. deliver the packets of the first flow in the set until all packets are sent;
7. remove this flow from the set;
8. **end while**
9. **return**

Figure. 1: The pseudocode of MCPF algorithm

From the perspective of probability density distribution graph, Eq. 1 can be thought to be the ratio of the area between B_i and B_i+Ct to the area where flow size is more than B_i. Combining with the characteristics of flow size distribution in data center networks, we can summarize the following conclusions:

- No matter long flows or short flows, the value of p_i will remarkably increase in a short time when the bytes they have sent approximate to a peak of flow size distribution, and then they are likely to be served in switches so as to finish transmission as soon as possible.
- The workloads in data center networks, e.g. web search and data mining, usually concentrate on the short flows, and the areas that short flows cover in a time interval are far larger than those of long flows. Thus, short flows are more likely to be scheduled.
- Once the size of a flow crosses over the peak of probability density distribution but this flow doesn't complete its transmission yet, the completion probability of this flow will sharply drop in the next period, and then it cannot compete over the new arrival flows or short flows.
- For two long flows whose arrival times are different, the denominator of the former flow in Eq. (1) is obviously smaller than that of the later flow, and their numerators are close, so the former flow is superior to the later flow. This result is propitious to decrease the average FCT.

In a word, MCPF strategy has positive effect on speeding up flow transmission, and we will testify this qualitative predication via a queueing theory based mathematical model in the next section.

3. Model and Analysis

To evaluate the performance of MCPF strategy, we employ queueing model to analyze the problem of flow scheduling in switches. We also compare MCPF with LFLP and FCFS, and describe the expression of average FCT for the three strategies under different traffic loads.

3.1 Queueing Model

Beside the notations defined in Section IV, we assume the arrival rate of flows conforms to Poisson distribution with parameter λ, and the processing capacity of each switch port is μ. And then the traffic load is

$$\rho = \lambda \int_1^{+\infty} xf(x)dx / \mu \tag{2}$$

Because switches forward packets by matching the destination IP address, we let μ be a constant, and switches can handle one packet in a unit time. Let the period of flow scheduling is $1/\mu$, and then flow scheduling is performed after each packet is forwarded.

To simplify the analysis on flow scheduling strategies, we ignore such complicated circumstances as packet loss and retransmission during packet forwarding. According to TCP congestion control mechanism, each flow complies with slow start, and additive-increase, multiplicative-decrease. Let the upper bound of sending window is *M* packets(The sending window in existing Linux operation system is 64KB by default. If MSS is 1500 bytes, the sending window is 44 packets.). For a flow with *x* packets ($x \le 2^{\lceil \log_2 M \rceil + 1} + M - 1$), this flow will experience $\lceil \log_2(x+1) \rceil$ round-trip times (RTTs) since starting to transmit. For a flow whose packet number is more than $2^{\lceil \log_2 M \rceil + 1} + M - 1$, the extra RTTs that a flow exceeds the upper bound of sending window is $\left\lceil (x - 2^{\lceil \log_2 M \rceil + 1} + M - 1)/M \right\rceil$.

Consequently, the RTTs that a flow with *x* packets will undergo to finish transmission are as follows.

$$N_{\mathrm{RTT}}(x) = \begin{cases} \lceil \log_2(x+1) \rceil, & x \le 2^{\lceil \log_2 M \rceil} + M - 1 \\ \left\lceil \dfrac{x - 2^{\lceil \log_2 M \rceil} + M - 1}{M} \right\rceil + \lceil \log_2 M \rceil + 1, & x > 2^{\lceil \log_2 M \rceil} + M - 1 \end{cases} \tag{3}$$

In FCFS strategy, all packets are located in the same queue, and the scheduling obeys the arrival order of packets. We assume the buffer of switches is infinite, and then the scheduling strategy can be deemed to follow the M/G/1 queueing model.

According to the results of M/G/1 queueing model and Pollaczek-Khintchine formula [7], the average queueing delay of each packet is

$$E[W] = \frac{\rho(1 + C_b^2)}{2(1-\rho)} \tag{4}$$

where C_b^2 is the variance of service time.

Since we assume μ is constant, $C_b^2 = 0$, and Eq. (4) can be transformed to

$$E[W] = \frac{\rho}{2(1-\rho)} \tag{5}$$

In LFLP and MCPF strategies, there are multiple queues with different priorities in switches. When switches schedule packets, they conform to strict priority, and only if all queues with higher priority are empty, the packets in a specific queue can be served. For simplifying the analysis on flow scheduling, we assume there are infinite queues in the input ports of switches, and each queue matches a flow with specific size. In addition, the buffer of each queue is infinite, and they can store all arrival packets. As the period time of flow scheduling is $1/\mu$, the order of packet forwarding is re-sorted in next period time, so the new coming packet with higher priority won't be served until

this cycle is over. Therefore, these two flow scheduling strategies can be abstracted as a M/G/1 queueing model with non-preemptive priority.

Based on the conclusions of M/G/1 queueing model with non-preemptive priority [8], the average queueing delay of packets in the queue with *l*th priority is

$$E[W_l] = \begin{cases} \dfrac{\rho}{2(1-\rho_1)}, & l = 1 \\ \dfrac{\rho}{2(1-\sum_{i=1}^{l}\rho_i)(1-\sum_{i=1}^{l-1}\rho_i)}, & l > 1 \end{cases} \tag{6}$$

where ρ_i is the traffic load in the queue with *i*th priority.

3.2 FCT Analysis

For FCFS strategy in a single queue, the number of RTT that a flow with *x* packets will suffer and the average queueing delay can be calculated via Eq. 3 and Eq. 5. Therefore, the average FCT of FCFS strategy is approximated to the following formula.

$$FCT_{\mathrm{TCP}} = \int_1^{+\infty} [\frac{\rho}{2(1-\rho)} + 1] N_{\mathrm{RTT}}(x) f(x) dx \tag{7}$$

For LFLP and MCPF flow scheduling strategies, there are infinite queues in each input port of switches, and these queues are respectively corresponding to the flows with different size. Notice that switches arrange the order of scheduling whenever a packet is forwarded, and the flow size is unknown for switches, switches will insert each packet of a flow into distinct queue.

The priorities of flows are gradually degraded with the increase of flow size in LFLP strategy, i.e. the first packet of each flow is placed in the queue with the highest priority, and the second packet of each flow is placed in another queue with the second highest priority, and so on. Therefore, for a flow with *x* packets, its packets will be inserted into the first *x* queues one by one. From the queue's point of view, only these flows whose packet number is more than *l* have packets in the queue with priority *l*. Because the arrival rate of flows follows Poisson distribution with parameter λ, and the flow size distribution is *f*(*x*), the arrival rate of packets in the *l*th queue roughly equals to the following formula.

$$\lambda_l = \lambda \int_l^{+\infty} f(x) dx \tag{8}$$

From Eq. 8 and the definition of traffic load in Eq. 2, we can derive the traffic load of the *l*th queue as

$$\rho_l = \frac{\rho \int_l^{+\infty} f(x) dx}{\int_1^{+\infty} x f(x) dx} \tag{9}$$

Since packet loss is neglected, and the packets in the queue with higher priority are certainly served before these packets in the queues with lower priorities, the problem of out-of-order is prevented. Assuming the sequence number of all flows starts from 1. According to Eq. 3 and Eq. 6, we have the average FCT of a flow with *x* packets in LFLP strategy is

$$FCT_{\mathrm{LFLP}}(x) = N_{\mathrm{RTT}}(x) + \frac{\rho}{2} \sum_{r=1}^{N_{\mathrm{RTT}}(x)} \frac{1}{(1-\sum_{i=1}^{L_r}\rho_i)(1-\sum_{i=1}^{L_r-1}\rho_i)} \tag{10}$$

where L_r means the maximum sequence number of sending window in the rth round transmission, and it can be expressed as follows.

$$L_r = \begin{cases} 2^r - 1, & r < \lceil \log_2 M \rceil + 1 \\ 2^{\lceil \log_2 M \rceil} + M - 1 + M(r - \lceil \log_2 M \rceil - 1), & \lceil \log_2 M \rceil + 1 \le r < N_{\mathrm{RTT}}(x) \\ x, & r = N_{\mathrm{RTT}}(x) \end{cases} \tag{11}$$

So the average FCT of LFLP strategy is

$$FCT_{\mathrm{LFLP}} = \int_1^{+\infty} FCT_{\mathrm{LFLP}}(x) f(x) dx \tag{12}$$

For MCPF strategy, the flow scheduling is based on the probability of flows finishing transmission in a period time rather than the flow size. So some flows which have sent a great number of packets may be scheduled before the non-long flows due to the number of packets being in a peak of flow size distribution. To get the average FCT value of MCPF strategy, we should recompose the ranking of each queue according to the probability density distribution function of flow size, and forward packets based on the following scheduling principle: If the completion probability of queue i computed by Eq. 1 is larger than that of queue j, the packets in queue i will be dequeued earlier than those packets in queue j.

Consequently, the computation process of FCT in MCPF strategy is similar to Eq. 12, but the FCT of a flow with x packets has a little difference. The detailed formula is

$$FCT_{\mathrm{MCPF}}(x) = N_{\mathrm{RTT}}(x) + \frac{\rho}{2} \sum_{r=1}^{N_{\mathrm{RTT}}(x)} \frac{1}{(1 - \sum_{i=1}^{L'_r} \rho'_i)(1 - \sum_{i=1}^{L'_r - 1} \rho'_i)} \tag{13}$$

where ρ'_i is the traffic load of the ith queue after arrangement, and L'_r is the queue with the lowest priority in the rth round transmission.

4. Performance Evaluation

4.1 Simulation Scenario

Platform: We implement the prototype of MCPF in Mininet [9]. Mininet is a Linux kernel based virtualization simulation platform, which have been widely used in performance evaluation of OpenFlow network, and it can construct a small-scale network in single computer. All switches and hosts in the virtual network share the CPU and memory of the computer, and realistic network traffic is injected into Mininet. Due to the limitation of hardware resources, the link bandwidth in the virtual network is restricted, and it often supports tens of Mbps, which corresponds to 1Gbps of practical link bandwidth in our simulation.

Topology: We use a 4-pod FatTree as the data center network topology, which contains 20 switches and 16 end hosts. The link bandwidth is 10Mbps, and the propagation delay of each link is 1ms. Each ToR switch connects to 2 hosts, while the buffer of each switch port can store 100 packets.

Workloads: Two empirical workloads are selected as our traffic load, which are web search workload [3] and data mining workload [5], respectively. Both of the flow size distributions of the two workloads exhibit remarkably heavy tail features, and their distributions are centralizing in short flows, while most of bytes are generated by long flows.

In our experiments, the communication pairs are randomly selected, and we use iperf traffic generator to emulate the flows of data center networks. The packet arrival rate follows Poisson distribution, and the rate is computed as the following formula according to Eq. 2.

$$\lambda = \rho\mu / \int_1^{+\infty} xf(x)dx \tag{13}$$

where ρ is traffic load, and μ is the link bandwidth.

When a flow is generated, the flow size is determined according to the probability density function, and then it continues to send until all bytes are transmitted.

4.2 Flow Scheduling Strategies for Comparison

FCFS: All packets are in the same queue, and scheduled by the order of packets arriving in switch ports. This strategy is selected as the baseline of our evaluation.

LFLP: There are 8 queues in each switch port, and the priorities of each queue are different. For a flow, its priority is decreasing with the raise of packets that have sent. When the packet number exceeds a demotion threshold, this flow degrades to a lower priority, i.e. the subsequent packets are inserted into another queue. The demotion threshold in LFLP is determined by the approach presented in [6]. The packets in the same queue is scheduled based on FCFS.

MCPF: The implementation follows the proposed method, and each switch port also have 8 queues. The completion probabilities are computed according to Eq. 1, and the flow size range for each queue is determined by the probability distribution and *k*-means clustering. We choose POX as the controller, and the information query is executed every 10s. Each queue schedules its packets by FCFS.

To fairly evaluate the performance of the three flow scheduling strategies, all end hosts employ standard TCP-New Reno as transport protocol in our experiments, and switches use DropTail queues. The initial sending windows of TCP is 12KB, while the maximum value is 64KB.

4.3 Experimental Results

To broadly evaluate the performance of MCPF, we collect the statistic of both average and 99-th percentile FCTs for short flows and long flows. Because the effects of flow scheduling strategies can be adequately reflected in heavy loads, we set the value of traffic loads from 0.5 to 0.95. The results of FCTs are shown in Fig. 2 and Fig. 3.

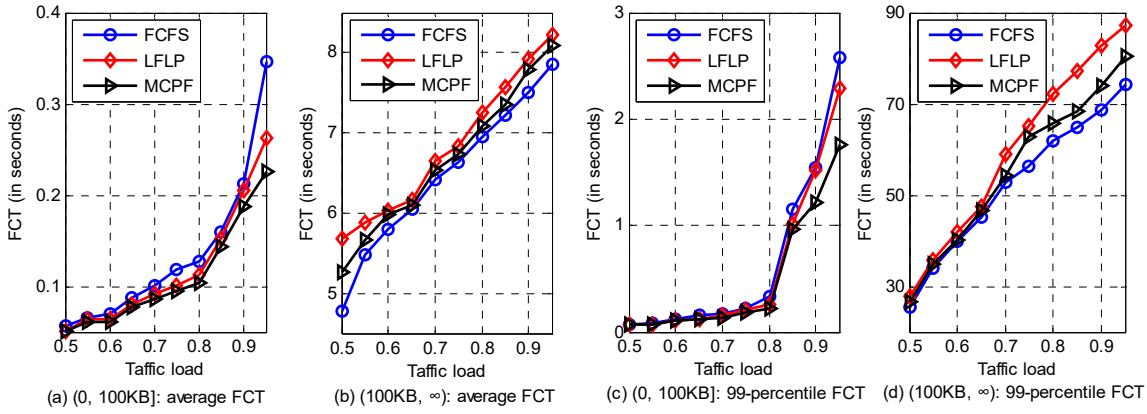

Figure 2. The FCTs of short flows and long flows under different traffic loads in web search workload

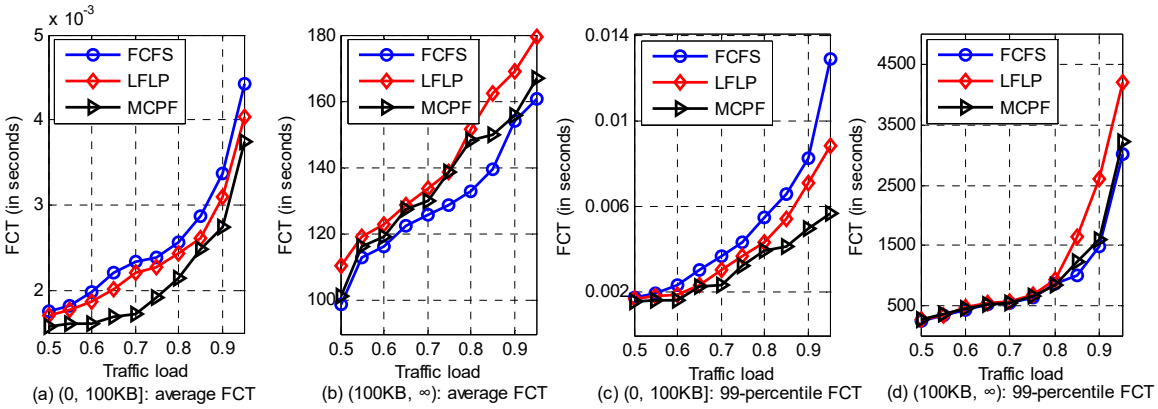

Figure 3. The FCTs of short flows and long flows under different traffic loads in data mining workload

The results of Fig. 2 and Fig. 3 demonstrate the advantage of MCPF compared to FCFS and LFLP, although the gap between MCPF/LFLP and FCFS is not as obvious as that in numerical simulation. The reason is that the queue number and buffer size in the mathematical model can be infinite, but it is impossible in reality, and thus the FCTs of flows in MCPF and LFLP also remarkably postpone along with the increase of traffic loads. However, no matter which workload is selected as the background flows, MCPF strategy always achieves the smallest FCT for short flows that are less than 100KB, which is even better than LFLP strategy. We think the relatively long polling period in practical networks makes the impact of new arrival flows rising. The gain of 99-percentile FCT is significantly salient, which indicates most of short flows can be completed quickly in MCPF. In addition, the results of short flows and long flows in data mining workload is polarized due to its traffic pattern, which is composed of these flows less than 3 packets and those ones larger than 35MB.

Another unsurprising phenomenon is the FCT of long flows in FCFS is shorter than that of LFLP and MCPF, and the transmission of long flows is seriously throttled in LFLP. We think it maybe result from the possible starvation of some long flows in the queues with low priority, and they have to wait for a long time to be scheduled. Moreover, due to the finite buffer in switch ports, packet loss or retransmission will happen, which also lengthens the FCT of long flows. Besides, frequent flow scheduling may also bring negative effect on the decrease of FCT of long flows.

5. Conclusions

Decreasing the FCT in information-agnostic data center networks is a challenging problem. In this paper, we present the design and implementation of MCPF, a flexible flow scheduling strategy to accelerate the transmission of these flows that are about to complete in a period time. A queueing theory based mathematical model is proposed to analyze the performance of MCPF and other two scheduling strategies, and the numerical simulation validates the analytical results. To implement MCPF in practical environment, we explore the challenges and corresponding solutions, and evaluate MCPF in Mininet. The experimental results demonstrate that MCPF achieves the best performance compared to LFLP and FCFS strategies.

Acknowledgements

This work is supported by the State Key Development Program for Basic Research of China under Grant No. 2012CB315806, the National Natural Science Foundation of China under Grant

No. 61379149, Jiangsu Province Natural Science Foundation of China under Grant Nos. BK20140068 and BK20140070.

References

[1] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowstron, *Better never than late: Meeting deadlines in datacenter networks,* In Proc. ACM SIGCOMM, 2011.

[2] D. Abts, B. Felderman, *A guided tour of data-center networking*, Commun. ACM, vol. 55, no. 6, pp. 44–51, June 2012.

[3] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, et al. *Data center TCP (DCTCP)*, in Proc. ACM SIGCOMM, 2010.

[4] T. Benson, A. Akella, and D. Maltz, *Network traffic characteristics of data centers in the wild*, In Proc. IMC, 2010.

[5] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, et al. *VL2: A scalable and flexible data center network*, In Proc. ACM SIGCOMM, 2009.

[6] W. Bai, L. Chen, K. Chen, D. Han, C, Tian, H. Wang, *Information-agnostic flow scheduling for commodity data centers,* In Proc. USENIX NSDI 2015.

[7] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. Fundamentals of Queueing Theory. Wiley-Interscience, 2008.

[8] C. Lu. Queueing Theory. Beijing University of Post and Telecommunication Press, 1993.

[9] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, *Reproducible network experiments using container-based emulation*, In Proc. ACM CoNEXT, 2012.

[10] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, *Improving datacenter performance and robustness with multipath tcp*, In Proc. ACM SIGCOMM, 2011.