P4INC-AOI: When In-Network Computing Meets All-Optical Interconnect for Adaptive and Low-Latency Optical DCN

Xuexia Xie[1], Hao Yang[1], and Zuqing Zhu[1]

1. University of Science and Technology of China, Hefei, Anhui 230027, China, Email: zqzhu@ieee.org

Abstract: We propose and experimentally demonstrate P4INC-AOI for orchestrating in-network computing and all-optical interconnect in an optical data-center network. Experimental results show that P4INC-AOI reduces job completion time of Hadoop MapReduce jobs by 77.9% on average.

OCIS codes: (060.1155) Software-defined optical networks; (060.4251) Networks, assignment and routing algorithms.

1. Introduction

Nowadays, the traffic related to data-centers (DCs) has become the largest contributor to Internet traffic [1], and their quality-of-service (QoS) demands have put great pressure on traditional DC networks (DCNs) based on electrical packet switching (EPS) [2]. For instance, distributed machine learning cannot run well in a DCN where there are inter-rack bottlenecks [3], and virtual reality cannot be supported without satisfying its low-latency requirement. Therefore, people are considering to introduce optical circuit switching (OCS) in DCNs [4], because it provides larger bandwidth capacity, shorter data transfer latency, and higher energy efficiency than EPS [5]. However, OCS can also make a DCN less flexible and more difficult to adapt to highly-dynamic inter-rack traffic, due to its larger switching granularity and longer reconfiguration latency [6]. Although the issues can be partially addressed by building hybrid optical/electrical DCNs (HOE-DCNs) [3] or increasing the connectivity of optical DCNs (O-DCNs) with multi-level designs [6], certain drawbacks (*e.g.*, complicated network architecture and management, and increased port count of top-of-rack (ToR) switches) cannot be avoided. This motivates us to study how to improve the adaptivity of O-DCNs with new techniques.

Recent studies on programmable data plane (PDP) [7] suggested that in addition to packet forwarding, a PDP switch can realize sophisticated packet processing, such as packet aggregation and arithmetic operations on fields, at line-rate [8]. Hence, instead of relying on the servers on end-hosts, computing jobs can be executed directly within the PDP switches that are already used to forward packets, which leads to the idea of "in-network computing" [9]. Since it terminates transactions as they traverse a network, in-network computing can achieve promising traffic reshaping and significantly reduce traffic load [9]. This makes in-network computing extremely useful for an O-DCN, because with proper orchestration, it can leverage PDP-based ToR switches to reshape/absorb bursty inter-rack traffic, and thus improve the QoS of network applications with minimized all-optical interconnect (AOI) reconfigurations.

Nevertheless, to the best of our knowledge, the orchestration of in-network computing and AOI in an O-DCN has not been studied in the literature yet. In this work, we propose and experimentally demonstrate *P4INC-AOI*, which is an inter-rack system for realizing highly-adaptive and low-latency O-DCNs. Specifically, we place a PDP switch at the ToR of each rack (namely, P-ToR switch), implement P4-based packet processing pipelines in it for in-network computing, and design a control plane system to orchestrate the in-network computing in P-ToR switches and the configuration of the AOI that connects them. In order to verify the performance of P4INC-AOI, we prototype it with off-the-shelf products (*i.e.*, a commercial 32×32 optical cross-connect (OXC), PDP switches based on Tofino chips, and commodity servers), build a small-scale but realistic O-DCN that consists of four racks, and run Hadoop MapReduce (H-M/R) jobs in it. Experimental results suggest that our P4INC-AOI can effectively absorb the inter-rack traffic caused by the H-M/R jobs, relieving the pressure of invoking AOI reconfigurations to adapt to dynamic network applications, and greatly accelerate the H-M/R jobs to reduce their job competition time by 77.9% on average.

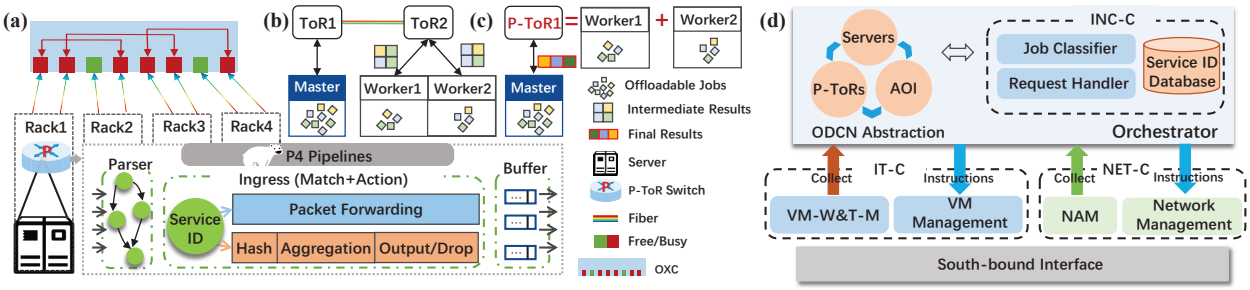

Fig. 1. System architecture and operation principle of P4INC-AOI, (a) Data plane, (b) and (c) Examples on running Hadoop MapReduce jobs without and with in-network computing in ToR switches, respectively, and (d) Control plane.

2. System Design and Operation Principle of P4INC-AOI

We show the system architecture and operation principle of P4INC-AOI in Fig. 1. The data plane in Fig. 1(a) follows the generic architecture of an O-DCN, where an AOI connects a set of server racks. However, we replace conventional ToR switches with P-ToR switches and program P4-based packet processing pipelines in them for in-network computing. Figs. 1(b) and 1(c) explain the executions of H-M/R jobs without and with in-network computing in ToR switches, respectively. In Fig. 1(b), as the workers and master of a H-M/R cluster run in different racks, the workers need to send their intermediate results to the master repeatedly and receive new computing jobs from it from time to time. This inevitably generates bursty inter-rack traffic and might lead to frequent reconfigurations of the AOI in an O-DCN. On the other hand, when in-network computing is in place (in Fig. 1(c)), we use a centralized orchestrator to activate related P4 pipelines on the master's P-ToR switch and replace the workers with them. Hence, the bursty computing jobs from the master can be finished without generating any inter-rack traffic. More importantly, as the in-network computing with P4 pipelines is hardware-based, it runs much faster than virtual machines (VMs) on servers. Therefore, both the inter-rack communication latency and computing time can be saved in P4INC-AOI.

Note that, due to the hardware restrictions of P-ToR switches (*i.e.*, their computing capability is still limited), we cannot offload arbitrary computing jobs to them, except for those that can be accomplished with data matching, hashing and aggregation and simple arithmetic operations. Fortunately, a few representative H-M/R jobs fall into this category, such as WordCount and TeraSort. For instance, WordCount counts the number of occurrences of each word in a large input set, which can be realized with a P4 pipeline that consists of data hashing and key-value stores. As shown in Fig. 1(a), each P-ToR switch distinguishes the packets of different applications based on their service ID. Specifically, we let the control plane assign a service ID to each application. Hence, a P-ToR switch will just forward the packets as usually, if their applications are not offloadable, and it will apply in-network computing to the others if necessary. Meanwhile, the design also enables the centralized orchestrator to activate/deactivate pre-programmed P4 pipelines in a P-ToR switch in runtime (*i.e.*, by assigning/reassigning the right service IDs to related packets dynamically).

Fig. 1(d) shows our design of the control plane for P4INC-AOI. The IT controller (IT-C) includes a VM workload and traffic monitor (VM-W&T-M) and a VM management module. The VM-W&T-M collects the status of the VMs running computing jobs, and the VM management implements the instructions from the orchestrator. The network controller (NET-C) uses a network abstraction module (NAM) to monitor P-ToR switches and the AOI and abstract the inter-rack topology of the O-DCN, while the network management module in it reconfigures the network elements according to the orchestrator's instructions. Note that, P-ToR switches can be managed with P4Runtime, and the optical switches in the AOI (*e.g.*, OXCs) can be controlled with OpenFlow 1.3. On top of the IT-C and NET-C, we design an orchestrator, which visualizes the O-DCN based on the reports from the IT-C and NET-C. Based on the O-DCN abstraction, the in-network computing controller (INC-C) activates/deactivates P4 pipelines in P-ToR switches to offload computing jobs from VMs, and instructs the IT-C and NET-C to reconfigure the O-DCN accordingly.

3. Experimental Demonstrations

To demonstrate the effectiveness of our P4INC-AOI, we prototype a small-scale O-DCN that consists of four racks, each of which uses a PDP switch based on Tofino chips as its P-ToR switch, and an AOI that is based on a commercial 32×32 OXC. The line-rate of the P-ToR switches is 40 Gbps. As for the computing jobs that need to be executed in the O-DCN, we consider the WordCount of H-M/R and program P4 pipelines in P-ToR switches to support the in-network computing for it. In the experiments, a H-M/R cluster processes $[0.45, 2.25]$ GByte data for each WordCount job.

We first conduct experiments to measure the "initial response time" of each WordCount job, which is defined as

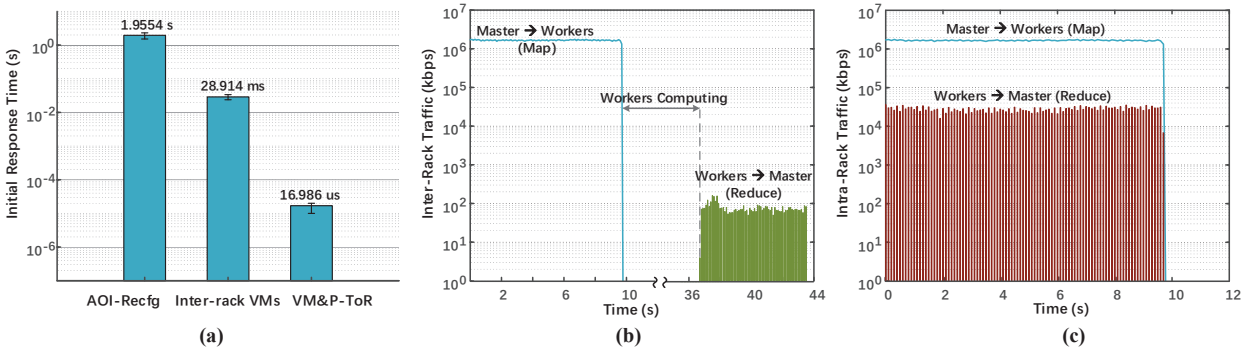

Fig. 2. Experimental results on (a) Initial response time, (b) Inter-rack traffic caused by a WordCount job in traditional O-DCN without in-network computing, and (c) Intra-rack traffic due to running a WordCount job in our P4INC-AOI.

the interval between when the master of the H-M/R cluster sends out the first packet and when it receives the first reply from the VM-based workers or a P-ToR switch. The experiments consider three scenario: 1) the master and workers are all VM-based, they are deployed on different racks, and an AOI reconfiguration needs to triggered to set up the connections among them (AOI-Recfg), 2) all the settings are the same as the first scenario, except for that the connections among the master and works can be set up without an AOI reconfiguration (Inter-rack-VMs), and 3) the master is still VM-based while the workers are realized with the P4 pipelines on the master's P-ToR switch (VM&P-ToR). Fig. 2(a) shows the experimental results, which indicate that with in-network computing, VM&P-ToR reduces the initial response time by 5 and 3 magnitudes over AOI-Recfg and Inter-rack-VMs, respectively. These results verify that compared with the conventional service provisioning scenarios in O-DCNs, our P4INC-AOI can greatly reduce the initial response time of offloadable computing jobs, which will be extremely useful to delay-sensitive applications.

Note that, P4INC-AOI actually has more advantages than just reducing the initial response time. Therefore, we conduct more experiments to demonstrate them. Specifically, we run WordCount jobs in both a traditional O-DCN without in-network computing and an O-DCN based on our P4INC-AOI. As for the traditional case, the H-M/R cluster consists of a master and 9 workers, where the workers are deployed in three racks and the master runs on a server in the fourth rack. Then, each WordCount job is executed according to the classic scheme in H-M/R: the master first distributes the job's data to the workers (*i.e.*, the Map stage), and then collects intermediate results from the workers and derives the final result based on them (*i.e.*, the Reduce stage). Fig. 2(b) plots the inter-rack traffic caused by a WordCount job running in the traditional O-DCN. It can be seen that in the Map stage, the master sends the job's data to the workers at a total bit-rate of $\sim$1.8 Gbps for around 9.8 seconds. Note that, the total bit-rate is set according to the data-receiving capability of each VM-based worker, even though the line-rate of each server's linecard is much higher. During the Map stage, the workers buffer and aggregate the data received from the master. Next, they perform the computing tasks required by the WordCount job, which takes $\sim$26.9 seconds, and during the period, the H-M/R cluster does not generate any inter-rack traffic, as shown in Fig. 2(b). Finally, the workers sends their intermediate results back to the master, which then concludes the WordCount job by outputting the final result. In Fig. 2(b), this Reduce stage takes $\sim$7.1 seconds, where the total data-rate from the workers to the master is much lower than that of the Map stage. This is actually expected, as the workers only need to send their counting results back.

We hope to point out that all the traffic plotted in Fig. 2(b) is for inter-rack communications, which not only consumes the throughput of the AOI but also can trigger AOI reconfigurations. For instance, when there are also other applications running in the O-DCN, we might need to reconfigure the AOI to set up the the connections among the master and works before the Map stage, or to tear down the connections during the silent period between the Map and Reduce stages for letting the traffic of other applications use the AOI. Nevertheless, the hassle can be avoided with our P4INC-AOI. To verify this, we run WordCount jobs by deploying a VM-based master and offloading the workers to the master's P-ToR switch. For fair comparisons, we still let the master send each job's data to the P-ToR switch at $\sim$1.8 Gbps, even though our P4 pipelines in it can process jobs at a much higher data-rate with in-network computing. Fig. 2(c) shows the traffic caused by a WordCount job running in an O-DCN based on our P4INC-AOI. As all the inter-rack traffic is avoided this time, we only plot the intra-rack traffic between the master's server and the P-ToR switch. We can see that the Map and Reduce stages of the WordCount job run in parallel because the P-ToR switch can process the job's data and return intermediate results instantly with its P4 pipelines. Therefore, the Reduce stage is finished almost the same time as the Map stage, which greatly reduces the job completion time (JCT). The experiments, which process different volumes of data in each WordCount job, suggest that comparing with the traditional O-DCN without in-network computing, our P4INC-AOI reduces the JCT of WordCount jobs by 77.9% on average.

4. Summary

We proposed P4INC-AOI to orchestrate in-network computing and AOI in an O-DCN. Experiments showed that our proposal effectively absorbed the inter-rack traffic caused by H-M/R jobs and reduced their JCT by 77.9% on average.

References

[1] Cisco Annual Internet Report (2018-2023), [Online]. Available: http://www.cisco.com/c/en/us/solutions/service-provider/ visual-networking-index-vni/index.html.
[2] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-data-center networks," *IEEE Netw.*, vol. 29, pp. 36-42, Sept./Oct. 2015.
[3] H. Yang *et al.*, "Which can accelerate distributed machine learning faster: hybrid optical/electrical or optical reconfigurable DCN?" *in Proc. of OFC 2022*, paper Th1G.5, Mar. 2022.
[4] C. Xie, "Datacenter optical interconnects: Requirements and challenges," *in Proc. of OI 2017*, pp. 37-38, Jun. 2017.
[5] Z. Zhu *et al.*, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15-22, Jan. 2013.
[6] K. Chen *et al.*, "Modular optical cross-connects (OXCs) for large-scale optical networks," *IEEE Photon. Technol. Lett.*, vol. 31, pp. 763-766, May 2019.
[7] P. Bosshart, *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87-95, Jul. 2014.
[8] A. Sapio *et al.*, "Scaling distributed machine learning with in-network aggregation," *in Proc. of NSDI 2021*, pp. 785-808, Apr. 2021.
[9] N. Zilberman, "In-network computing," Apr. 2019. [Online]. Available: https://www.sigarch.org/in-network-computing-draft/.