IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON KEY TECHNOLOGIES FOR SMART FACTORY OF INDUSTRY 4.0

Received September 29, 2017, accepted November 1, 2017, date of publication December 4, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2779462

Task Placement Across Multiple Public Clouds With Deadline Constraints for Smart Factory

BOYU LI[1], ZHIPENG ZHAO[1], YAN GUAN[1], NING AI[1], XIAOWEN DONG[2], AND BIN WU[1], (Member, IEEE)

[1]School of Computer Science and Technology, Tianjin University, Tianjin 30000, China
[2]DC Technology Laboratory, Huawei Technologies Company Ltd., Shenzhen 518000, China

Corresponding author: Bin Wu (binw@tju.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2016YFB0201403, in part by the Natural Science Fund of China under Grant 61372085, in part by the Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, Tianjin, China.

ABSTRACT The smart factory of Industry 4.0 has been regarded as a solution for handling the increasing production complexity caused by growing global economy and demand for customized products. Besides, it will make the interactions between humans, machines, and products become a highly competitive area for market capitalization in the near feature. Nowadays, cloud computing with the high performance of computing and self-service access plays an important role in realizing smart factor. To minimize the overall cost of company in a heterogeneous cloud environment, including multiple public clouds, while ensuring a proper level of quality-of-service, task placement across multiple public clouds is a critical problem, where task deadlines and long-haul data transmission costs between smart factory and different clouds must be considered. We formulate this task placement problem as an integer linear program (ILP) to minimize company cost under the task deadline constraint. With extensive simulations, we evaluate the performance of our proposed ILP model in heterogeneous public clouds with finite and infinite resources.

INDEX TERMS Cloud computing, heterogeneous cloud, integral linear programming (ILP), task placement.

I. INTRODUCTION

Automation and information systems such as enterprise resource planning and manufacturing execution system make factory productivity improve significantly. However, the current industrial production faces many critical challenges, such as environmental pollution, energy consumption and ever-shrinking workforce supply. Therefore, industrial processes need to achieve high flexibility and efficiency as well as low energy consumption and cost [1]. A strategic initiative called ''Industry 4.0'' proposed and adopted by the German government has already been proposed aiming to overcome the drawbacks of the current production lines [2]–[4]. Other main industrial countries have also been proposed similar strategies, taken, ''Industrial Internet'' by USA and ''Internet +'' by China for example. The Industry 4.0 describes a cyber-physical system (CPS) oriented production system that integrates production facilities, warehousing systems, logistics, and even social requirements to establish the global value creation networks [5].

The smart factory is an important feature of Industry 4.0 that addresses the vertical integration and networked manufacturing systems for smart production [2]. For smart factory to be implemented, it needs to process large amounts of data and large-scale calculation. That is why cloud computing is taken as the key technology to achieve the promising prospects of smart factory.

Cloud computing enables the smart factory to utilize elastic resources, such as servers, storage, software and so on over the Internet, as a utility – just like electricity – rather than having to build and maintain their own computing infrastructures [6]–[10]. Nowadays, multiple service providers can offer cheap public clouds with high performance and availability. Examples include Amazon, Google App Engine and Microsoft Azure. Smart factories are generally built on those diversified public clouds to meet various needs.

Generally, the most frequently processed work-flows on cloud computing are for commercial manufactures and business managements, etc., which can be considered as a multi-task project. Therefore, an important issue is how to rent public clouds to carry out a multi-task project in a cost-efficient manner, especially under task deadline constraints and heterogeneous cloud environments. Multiple

2169-3536 © 2017 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

public cloud providers create a heterogeneous cloud environment, in which different types of instances are offered with various prices and performance. This leads to a complex optimization problem: how to properly combine those instances from different providers, such that the total cost can be minimized while all computational tasks can be completed across heterogeneous clouds within the given deadlines. Besides, data transmission costs between the smart factory and clouds must be taken into account as well.

The above problem can be taken as a task placement problem. To come up with an optimal solution, the smart factory needs to consider the following factors: (1) what types of instances should be rented and from which providers; (2) how much amounts of the instances should be rented and how should the tasks be placed on them; and (3) what is the optimal tradeoff among the cost of the smart factory and service performance to meet the desired task deadline constraints while minimizing the overall cost.

Several existing works have studied the task placement problem in different ways. Gu *et al.* [11] exploit the dynamic frequency scaling technique and formulate an optimization problem to minimize cost, while guaranteeing the expected response time as the quality-of-service metric. LaCurts *et al.* [12] show that company can achieve a good task placement by understanding the interplay of underlying clouds and task demands. Shi *et al.* [13] focus on the problem of scheduling embarrassingly parallel jobs composed of a set of independent tasks and consider energy consumption during scheduling. The objective is to determine both a task placement plan and a resource allocation plan for jobs to minimize the Job Completion Time (JCT). In [14] a rule based task scheduling method is presented for allocating tasks to time slots of rented Virtual Machines (VMs) with a task right shifting operation and a weighted priority composite rule. A Unit-aware Rule-based Heuristic (URH) is proposed for elastically provisioning VMs to task-batch based work-flows to minimize the rental cost in DAG-based platforms (such as Dryad, Spark and Pegasus). Pham *et al.* [15] introduce a novel two-stage machine learning approach for predicting workflow task execution times for varying input data in the cloud. In order to achieve high accuracy predictions, their approach relies on parameters reflecting runtime information and two stages of predictions.

To improve the energy efficiency of heterogeneous servers in the cloud computing system, Zhang *et al.* puts forward a non-cooperative game based task scheduling and computing resource allocation algorithm [16]. They first use a non-cooperative game to model the task scheduling and computing resource allocation process of the servers, and the server's utility function is modeled as the unit power efficiency. Then, they prove the existence of a Nash Equilibrium point of the game, and use a Lagrange multiplier-based distributed iteration algorithm to solve the game. By considering task placement for both elastic and inelastic tasks, the paper [17] develops a resource management and allocation framework to reduce energy consumption of datacenters.

Nevertheless, the above existing works only consider task placement inside a private cloud, or between private cloud and a single public cloud. Our work differs from them by taking into account multiple public clouds in a heterogeneous environment, as well as the long-haul transmission costs between the smart factory and the public clouds. In particular, we formulate an Integer Linear Program (ILP) to solve the task placement problem and minimize the overall cost under the task deadline constraints.

Our main contributions include:

(1) We formulate the task placement problem in heterogeneous cloud environment to minimize the cost of the smart factory under the task deadline constraint, where an ILP approach is adopted and transmission costs between the smart factory and public clouds are considered.

(2) We evaluate the proposed model under scenarios involving finite and infinite public cloud resources.

The rest of the paper is organized as follows: Section II describes network model (including models for tasks and multiple public clouds). Section III formulates task placement into an ILP problem. Numerical results are presented in Section IV and we conclude the paper in Section V.

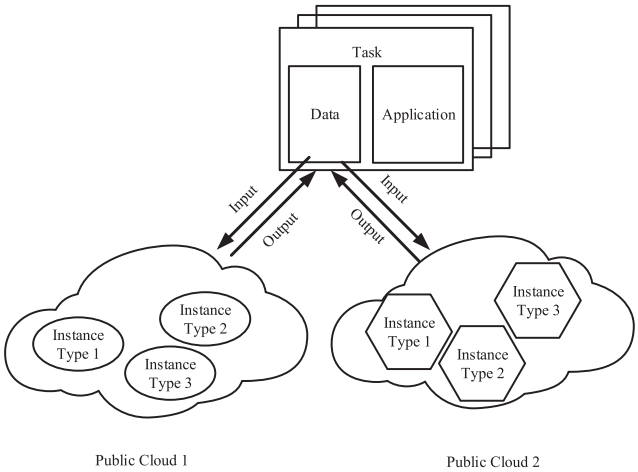


FIGURE 1. Task placement in heterogeneous clouds.

II. MODEL

A. MULTIPLE PUBLIC CLOUD MODEL

As shown in FIGURE 1, each public cloud offers multiple types of instance with different performance and prices. Cloud providers charge the company for each renting instance on an hourly basis. Nowadays, to assess the performance of an instance, application-specific benchmarks will be run on it, or to use some publicly available cloud benchmarking services, such as CloudHarmony [18]. CloudHarmony defines performance of cloud instances in the units named CloudHarmony Compute Units (CCU).

Furthermore, we assume two scenarios: infinite and finite public clouds. In the infinite scenario, the number of renting instance is not limited. While in the finite scenario, the public clouds impose constraints on the number of renting instances.

B. TASK MODEL

We assume a multi-task project with independent tasks. Task computation time, as measured by the running time on 1 CCU instance without interruption, are assumed to be known and constant. Generally, the deadline for tasks in one multi-task project is the same. These tasks will be processed in a heterogeneous cloud environment consisting of multiple public clouds, as shown in FIGURE 1. Each task can only be computed on one instance. Multiple tasks are processed on the instance in a sequential manner as long as the deadline constraint is not violated, and only one task can be processed at the same time. It is notable that the order of tasks to be processed in one instance is not important, because tasks are independent from each other.

In our model, we also assume that each task needs to process a certain amount of data that is stored on disk of smart factory and the smart factory needs to pay for data transmission from it to public clouds, whereas there is no data transmission among tasks. For data-intensive tasks, transmission cost may significantly contribute to the overall cost [19]–[21]. We assume that data transmission rate from the company to each public cloud is the same but the cost is different.

III. PROBLEM FORMULATION

Based on the task and cloud models in Section II, in this section we formulate the task placement problem into an optimal ILP. It minimizes the total cost for placing the given multi-task project across multiple public clouds by jointly considering individual instance cost and performance, as well as task deadline and data transmission cost.

A. NOTATION LIST

Parameters describing tasks:

I: The total number of all tasks.

s_i: The i-th tasks in multi-task project S, where $S = \{s_1, \cdots s_i, \cdots s_I\}$.

d_i: The size of data needed to transmission for task s_i.

a_i: The computation time of task s_i

t : the deadline for all tasks in one multi-task project.

Parameters describing public clouds:

M: The total number of candidate public clouds.

N_m: The number of instance types belonging to the m-th public cloud, where $m \in \{1, 2, \cdots M\}$.

K_{mn}: The number of n-th type instances belonging to the m-th public cloud, where $n \in \{1, 2, \cdots N_m\}$, $m \in \{1, 2, \cdots M\}$.

Parameters describing instance type :

cv_{mn}: The cost for renting n-th type instance belonging to the m-th public cloud per hour..

CCU_{mn}: The performance of n-th type instances belonging to the m-th public cloud in CloudHarmony Compute Units (CCU).

Parameters describing data transmissions:

cd_m: The cost of data transmission between smart factory and m-th public cloud per MB.

x_{imnk}: Binary variable. It takes 1 if s_i is put on the i-th task on the k-th instance of the n-th type belong to the m-th public cloud, and 0 otherwise.

B. ILP FORMULATION

$$\text{Minimize} \left[\sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{k=1}^{K_{mn}} \left(\left\lceil \frac{x_{imnk} a_i}{CCU_{mn}} \right\rceil cv_{mn} + x_{imnk} d_i cd_m \right) \right] \tag{1}$$

$$s.t. \sum_{i=1}^{I} x_{imnk} a_i < tCCU_{mn}, \quad \forall m \in \{1, 2, \cdots, M\}, \quad n \in \{1, 2, \cdots, N_m\}, \quad k \in \{1, 2, \cdots, K_{mn}\} \tag{2}$$

$$\sum_{m=1}^{M} \sum_{n=1}^{N_m} \sum_{k=1}^{K_{mn}} x_{imnk} = 1, \quad \forall i \in \{1, 2, \cdots, I\} \tag{3}$$

$$\frac{x_{imnk} a_i}{CCU_{\text{mn}}} \leq t, \quad \forall i \in \{1, 2, \cdots, I\} \tag{4}$$

Objective (1) minimizes the total cost. The first term is the cost of renting the instances, and the second term is the cost of all data transmissions between the smart factory and the public clouds. Which tasks can be put in one instance is constrained by (2). Constraint (3) says that each task can be placed onto at most one instance. Constraint (4) ensures that all tasks can be completed before deadline. In the finite public cloud scenario, the number of rented instance for each type constrained by (5).

$$\sum_{k=1}^{K} x_{imnk} \leq N_0, \quad \forall i \in \{1, 2 \cdots I\}, \quad \forall m \in \{1, 2 \cdots M\}, \ \forall n \in \{1, 2 \cdots N_m\} \tag{5}$$

TABLE 1. Instance parameters.

Public Cloud	Instance Type	Performance/CCU	Price/$/h
Public Cloud$_1$	v_{11}	0.8	0.8
	v_{12}	1.1	1.2
	v_{13}	1.8	0.2
Public Cloud$_2$	v_{21}	0.8	0.9
	v_{22}	1.6	1.8
	v_{23}	0.0	2.3

IV. NUMERICAL RESULTS AND DISCUSSIONS

We set simulation parameters on instances as in table 1. Unlimited number of rented instances is assumed in the infinite public cloud scenario. In contrast, we assume at most 10 rented instances for each type in the finite public cloud scenario.

A. THE RELATIONSHIP BETWEEN COST AND COMPUTATION TIME

In this experiment we assume three kinds of multi-task project. The computation time of task which is contained each multi-task project follows Gaussian distribution $N(0.2, 0.05)$, $N(0.5, 0.05)$ and $N(1, 0.05)$, respectively.

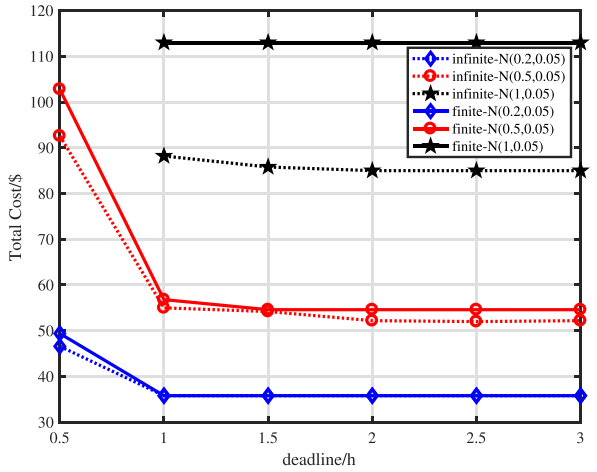


FIGURE 2. Relationship between cost and computation time.

Obviously, the mean of computation of time of above three project is equal to the mean of Gaussian distribution and increasing. The total number of tasks belonging to each multi-task project is 50 and the size of transmitted data for each task is the same 25 MB.

We simulate above three multi-task projects in infinite and finite public cloud scenarios and the results are distinguished by different line types shown in the legend of FIGURE 2. FIGURE 2 shows that the total cost always decreases with the deadline. This is because more tasks can be placed on one instance for a large deadline, and thus decreases the required total number of instances for all tasks. Constraint (2) plays an important role on this. We can also see that the total cost increases with the mean of computation time increasing under the same deadline. This is due to constraint (3). The reason is that if the mean computation time increases, we have to rent high performance instances with a higher price to complete all tasks in time, which will increase the total cost. Note that there is no solution for the multi-task project with computation time following $N(1, 0.05)$ in finite and infinite public cloud scenarios when the deadline is 0.5 h. This is because that those tasks in the above multi-task project can't be finished even we rent highest performance instance under the deadline is 0.5 h. This is caused by constraint (4).

FIGURE 2 also shows that the total cost in the finite public cloud scenario is much higher than that in the infinite case. This is because the number of low price instances is limited by constraint (5). Therefore, high price instances must be rented to complete the tasks under the deadline constraint.

B. THE RELATIONSHIP BETWEEN COST AND THE SIZE OF DATA TRANSMITTED

In this experiment, we also assume three kinds of multi-task project. The size of data to be transmitted for each task which is contained in every multi-task project follows satisfies Gaussian distributions $N(25, 0.05)$, $N(50, 0.05)$ and $N(75, 0.05)$ respectively. Obviously, the size of data to be transferred is increasing. The total number of tasks of each multi-task project is 50, and the computation time of each task in all multi-task projects is the same which is 0.2h on 1 CCU instance.

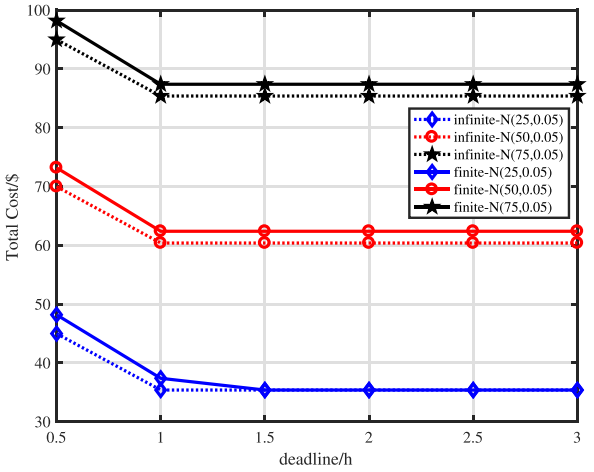


FIGURE 3. Relationship between cost and data transmissions.

We also simulate the above three multi-task projects in infinite and finite public cloud scenarios. Simulation results are shown in FIGURE 3. The legend in FIGURE 3 is similar to that in FIGURE 2. We can see that the total cost increases with the size of the transmitted data growing. The second part of the objective function plays an important role on this. FIGURE 3 also shows that the total cost in the finite public cloud scenario is much higher than that in the infinite case. Again, this is because high price instances must be rented to complete the tasks under the deadline constraint.

V. CONCLUSION

We studied the task placement problem under the task deadline constraint in a heterogeneous cloud environment containing multiple public clouds. An Integer Linear Program (ILP) was formulated to solve this problem for total cost minimization. The proposed ILP considers not only the cost of renting various types of instances with distinct costs and performance in different clouds, but also the transmission cost between the company and the public clouds. Numerical results showed that the total cost increases with the computation time and the size of the transmitted data. Our results can be applied to both computation and data intensive tasks.

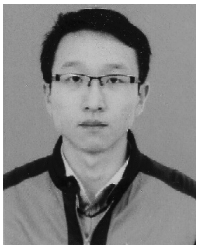
REFERENCES

[1] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination," *Comput. Netw.*, vol. 101, pp. 158–168, Jun. 2016.

[2] K. Henning, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," Working Group, Acatech—Nat. Acad. Sci. Eng., Munich, Germany, Final Rep. Industry 4.0, 2013.

[3] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2D big data: Content deliveries over wireless device-to-device sharing in realistic large scale mobile networks," *IEEE Wireless Commun.*, to be published.

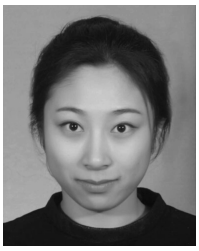
[4] S. Wang, Y. Zhang, H. Wang, Z. Wang, X. Wang, and T. Jiang, "Large scale measurement and analytics on social groups of device-to-device sharing in mobile social networks," *Springer Mobile Netw. Appl.*, vol. 23, no. 4, pp. 1–13, 2017.
[5] S. Wang *et al.*, "Implementing smart factory of industrie 4.0: An outlook," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 1, p. 3159805, 2016.
[6] X. Wang, Z. Sheng, S. Yang, and V. C. M. Leung, "Tag-assisted social-aware opportunistic device-to-device sharing for traffic offloading in mobile social networks," *IEEE Wireless Commun. Mag.*, vol. 23, no. 4, pp. 60–67, Aug. 2016.
[7] Z. H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Comput. Surv.*, vol. 47, no. 4, p. 63, 2015.
[8] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *J. Supercomput. J. Supercomput.*, vol. 54, no. 2, pp. 252–269, 2010.
[9] A. Behera, B. K. Ratha, and S. Sethi, "Green cloud computing: A survey," *Int. J. Sci. Eng. Adv. Technol.*, vol. 4, no. 12, pp. 763–767, 2017.
[10] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A survey on resiliency techniques in cloud computing infrastructures and applications," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2244–2281, 3rd Quart., 2016.
[11] L. Gu, D. Zeng, A. Barnawi, S. Guo, and I. Stojmenovic, "Optimal task placement with QoS constraints in geo-distributed data centers using DVFS," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 2049–2059, Jul. 2015.
[12] K. LaCurts, S. Deng, A. Goyal, and H. Balakrishnan, "Choreo: Network-aware task placement for cloud applications," in *Proc. ACM*, 2013, pp. 191–204.
[13] L. Shi, Z. Zhang, and T. Robertazzi, "Energy-aware scheduling of embarrassingly parallel jobs and resource allocation in cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 6, pp. 1607–1620, Jun. 2017.
[14] Z. Cai, X. Li, and R. Ruiz, "Resource provisioning for task-batch based workflows with deadlines in public clouds," *IEEE Trans. Cloud Comput.*, to be published.
[15] T. P. Pham, J. J. Durillo, and T. Fahringer, "Predicting workflow task execution time in the cloud using a two-stage machine learning approach," *IEEE Trans. Cloud Comput.*, to be published.
[16] L. Zhang and J.-H. Zhou, "Task scheduling and resource allocation algorithm in cloud computing system based on non-cooperative game," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2017, pp. 254–259.
[17] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Online assignment and placement of cloud task requests with heterogeneous requirements," in *Proc. GLOBECOM*, Dec. 2015, pp. 1–6.
[18] M. Malawski, K. Figiela, and J. Nabrzyski, "Cost minimization for computational applications on hybrid cloud infrastructures," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1786–1794, 2013.
[19] M. Malawski, J. Meizner, M. Bubak, and P. Gepner, "Component approach to computational applications on clouds," *Procedia Comput. Sci.*, vol. 4, pp. 432–441, Jan. 2011.
[20] B. Wang, J. Jiang, and G. Yang, "ActCap: Accelerating MapReduce on heterogeneous clusters with capability-aware data placement," in *Proc. INFOCOM*, Apr. 2015, pp. 1328–1336.
[21] M. Malawski, T. Gubała, and M. Bubak, "Component-based approach for programming and running scientific applications on grids and clouds," *Int. J. High Perform. Comput. Appl.*. vol. 26, no. 3, pp. 275–295, 2012.



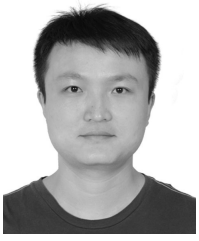
BOYU LI received the master's degree from the Key Laboratory of Wireless Sensor Networks, Yunnan Minzu University, Kunming, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Tianjin University, Tianjin, China, under the supervision of Prof. B. Wu. His research interests include cloud computing, computer systems and networking, wireless sensor networks, and compressed sensing.

ZHIPENG ZHAO is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Tianjin University, Tianjin, China, under the supervision of Prof. B. Wu. Her research interests include computer system and networking, optical communications and networking, and distributed algorithm.

YAN GUAN received the B.Eng. degree from Tianjin Polytechnic University, Tianjin, China, in 2016. She is currently pursuing the M.D. degree in computer systems and networking at Tianjin University, Tianjin, under the supervision of Prof. B. Wu. Her research interests include computer systems and networking, network survivability, and security issues.

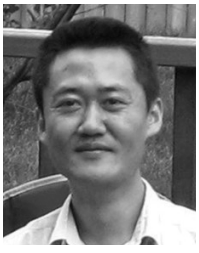


NING AI received the bachelor's degree in computer science and technology from Northeast Normal University, Jilin, China, in 2012. She is currently pursuing the master's degree in computer systems and networking at Tianjin University, Tianjin, China, under the supervision of Prof. B. Wu. Her research interests include computer systems and networking, optical and wireless communications and networking, network survivability, and security issues.



XIAOWEN DONG received the B.E. degree in electronic engineering from Southwest Jiaotong University, Chengdu, China, in 2005, the M.E. degree (Hons.) in electronic engineering from the National University of Ireland, Maynooth, Ireland, in 2008, and the Ph.D. degree in green optical networks from the University of Leeds, Leeds, U.K., in 2013. From 2005 to 2007, he was a Wireless Communication System Engineer with the Wuhan Research Institute, Wuhan, China. He is currently a Senior Research Engineer with the DC Technology Laboratory, Huawei Technologies Company, Ltd. His research interests include energy aware optical networks, novel data center architectures, and AI computing platforms.

He received the Cater Prize (the Best Ph.D. Thesis Award) in 2013, the Premium Award for Best Paper IET Optoelectronics in 2016, and the Cater Prize (the Best Ph.D. Thesis Award).



BIN WU received the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Hong Kong, in 2007. He was a Post-Doctoral Research Fellow with the Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, Canada, from 2007 to 2012. He is currently a Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His research interests include computer systems and networking and communications.