TAF
Journal of Advances in Technology and Engineering Research
JATER
2019, 5(1): 56-61

PRIMARY RESEARCH

SCCS: Streaming cooperative computing system for edge environment

Jiachen Feng [1*], Haiquan Wang [2], Shimin Wu [3], Yuhang Ma [4], Jiawei Guo [5]
[1, 2, 3, 4, 5] College of Software, Beihang University, Beijing, China

Keywords
Edge computing
Streaming data computing
Cooperative computing
Data privacy

Received: 8 December 2018
Accepted: 14 January 2019
Published: 20 February 2019

Abstract
With the development and popularity of smart devices, streaming data is continually generated at the network's edge. Under those edge scenarios, it takes high latency to process the streaming data by cloud computing. And uploading the raw data to cloud servers by networks faces data privacy issues. Streaming edge computing provides an opportunity to fill the gap between the real-time process and data privacy protection by conducting the major computing at the edge of networks where the streaming data is generated. In the paper, we propose and build SCCS, a Streaming data Cooperative Computing System suitable for an edge environment. This system (i) cooperatively conducts data analysis tasks on several servers at the edge of network topology; (ii) utilizes the geographically distributed data set; and (iii) protects data privacy and data value. A case study of analyzing real shared bicycle distribution demonstrates how edge nodes servers cooperate with others to process streaming data on real-time.

I. INTRODUCTION

Due to the explosive growth of data producers and consumers, next generation computing paradigm will be shifted in order to solve the processing and storing issues caused by the rapid increase of big data. At present centralized computing paradigm such as cloud computing could provide the computation and storage resources, which has been wildly used in big data business system. Analyzing and processing the streaming data in cloud computing environment, the common approach is to upload full amount raw data to the cloud data center, which entails transmission over networks, and conduct the computing task using the cloud resources. However, this kind of computing paradigm faces two limitations: (i) large amount of bandwidth is need to upload the burst in the amount of data generated at the network edge by an increasing number of connected devices, which may cause high latency; (ii) with the continuous refinement of data granularity and increment of data value, data owners have business interests and data privacy concerns and refuse to implement real-time streaming data processing by uploading the full amount raw data.

Not only cloud computing can process streaming big data. Decentralized computing paradigm has stepped into researchers' views, which is represent by fog computing [1, 2] and edge computing [3, 4]. A fog layer consisting of fog nodes is set between the cloud and the edge in fog computing environment. The fog nodes can complete most computing tasks instead of the cloud nodes and make computing power closer to data set. Nearby services are provided to end users according to the topology of edge networks which data privacy issues is partly protected by uploading less raw data on networks. Then researchers proposed a new computational model called edge computing, which consist of the cloud nodes and the edge nodes. The edge nodes where the data sets are stored conduct the computing tasks under the supervising and scheduling of the cloud nodes. In edge computing environment, raw data is processed locally which makes full use of resource in edge nodes. And less raw data transmitted over the network means less risk of data privacy leakage.

*Corresponding author: Jiachen Feng
†email: fengjiachen@buaa.edu.cn

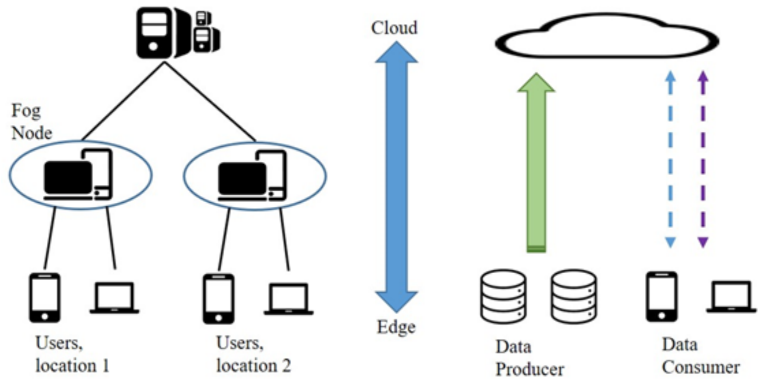

Fig. 1. Architecture of fog computing and cloud computing

In edge computing environment, analyzing streaming data generated and stored dispersedly faces the challenge of collectively conducting the computing on several edge nodes. As a key technology in edge computing, cooperative computing consists of task decomposition and result combination. In cloud computing environment, the full amount of raw data and the computing power needed in processing task are in cloud servers, and only task decomposition oriented to work flow needed to be concerned. However, because data set and computing resource are in several edge nodes of network topology, not only data process but also the data distribution must be considered.

In this paper, we propose Streaming Cooperative Computing System for Edge Environment (SCCS) to address the problem of processing the streaming data on geographically distributed edge nodes collectively. SCCS including central nodes and edge nodes conducts the computing task collectively by migrating the tasks to edge nodes near the streaming data source, which make it possible to analyzing the streaming data over edge nodes on real time. Without uploading the raw streaming data, our system can protect both data value and data privacy. Central node decomposes the streaming processing task according to the topology of data source recorded and schedules edge nodes collectively. Edge node analyses the streaming data and returns the result or mid-result to central nodes. Programmers can develop a streaming processing application, regardless of the number of data sources and its geographical distributions. In the paper, we focus on the build of SCCS, the represent and decomposition of service and the approach of analyzing steam data collectively. A real case study of analyzing the distribution of shared bicycles on real time demonstrates our system could process streaming data in edge computing environment.

The rest of this paper is organized as follows. The next Section reviews the existing works on computing paradigms, cooperative computing and streaming data processing. Section 3 proposes the concept and realization of our system. A real case study is given in Section 4. And Section 5 concludes the paper and presents our future work directions.

II. RELATED WORK

From the view of the topology, the development trend of the state-of the-art computing paradigm can be divided into centralization and decentralization. Cloud computing, which is on behalf of centralization computing paradigm, can provide the computation and the storage resource on demand as well as the software and the hardware resource needed in the computing process [5, 6]. Under the cloud computing environment, the full amount of raw data must be uploaded to cloud database, which faces the safety challenge of data privacy issues and business data disclosure [7, 8]. And single approach such as data encryption is not enough for protecting the data privacy [9]. As decentralization computing paradigm, the most representative model is fog computing [10] and edge computing [3]. Fog computing as an extension of cloud computing, a fog layer which can provide computing resource and services is set between the cloud and the end users [1]. Both cloud computing and fog computing can provide data storage, computing power and services while resources in fog computing locates closer to end users. [11] stated transmitting encrypted data makes it unable for fog devices to read privacy data, but does not guarantee that the Fog device transmits the correct report to the other gateways. Special demands like low latency and data privacy can't be met by cloud computing and fog computing, [7] proposes the edge computing. In edge computing environment, edge nodes have the ability of analyzing and storing data like cloud nodes and the computing over raw data is conducted at the edge nodes. Higher autonomy and processing raw data locally promises the privacy and the commercial value in the data. Both in academia and in-

ISSN: 2414-4592
DOI: 10.20474/jater-5.1.5

dustry, edge computing paradigm has been used in mobile edge computing [12] and smart city [13]. And with the development of 5G and IoT, the main feature of MEC is to push mobile computing, network control and storage to the network edges [14].

In edge environment, data is distributed in several edge nodes and multiple edge nodes participate in the one data processing task collectively. [15] propose a distributed cooperative computing model for embedded system. And the model can schedule the distributed dynamic execution units to compute collectively. Researchers propose to implement cooperative computing using the approach based on web service composition. [16] propose complicated abstract work flow can be executed by calling several web services and modeling method and service composition method are given. Researchers also try approaches based on graph or machine learning in order to provide more efficient service composition methods [17, 18]. But the aforementioned methods are oriented to service composition based on work flow, issues raised by data distributed over edge nodes can't be solved.

The exploitation of big data in edge environment has coursed the streaming data is generated from cloud server cluster to end devices. And the locations to cache or process streaming data also shift from cloud to edge [19]. Researchers propose a streaming data processing system including a data center and several edge data center [20], which is different from the traditional way of uploading full streaming data to the cloud. The latency is reduced by processing streaming data in the edge data center. In this situation streaming data is not processed in where it is generated, data privacy may be lacked even if it is only transmitted in the edge networks. It is common for researchers to choose Strom as data processing framework, which is an open source real-time streaming data processing system which is designed for distributed environment [21]. With the feature of low latency, fault-tolerance, horizontal scaling and parallel computations, Storm is used in business systems such as [22].

III. PROPOSED SYSTEM

Central node and edge node are the core parts of our system, which are geographically distributed and connected by networks. To meet the computation and storage demand, a node can be a single server or a cluster of servers. Central node saves the registration information of data and servers in the whole system. When a data analyzing task need to be done over multiple edge nodes, central node decomposes it into serval subtasks according to the registration information. Edge node conducts the subtask which only the local streaming data and computing are used. The coordination work between edge nodes is scheduled by central node in an approach of task decomposition and result composition.

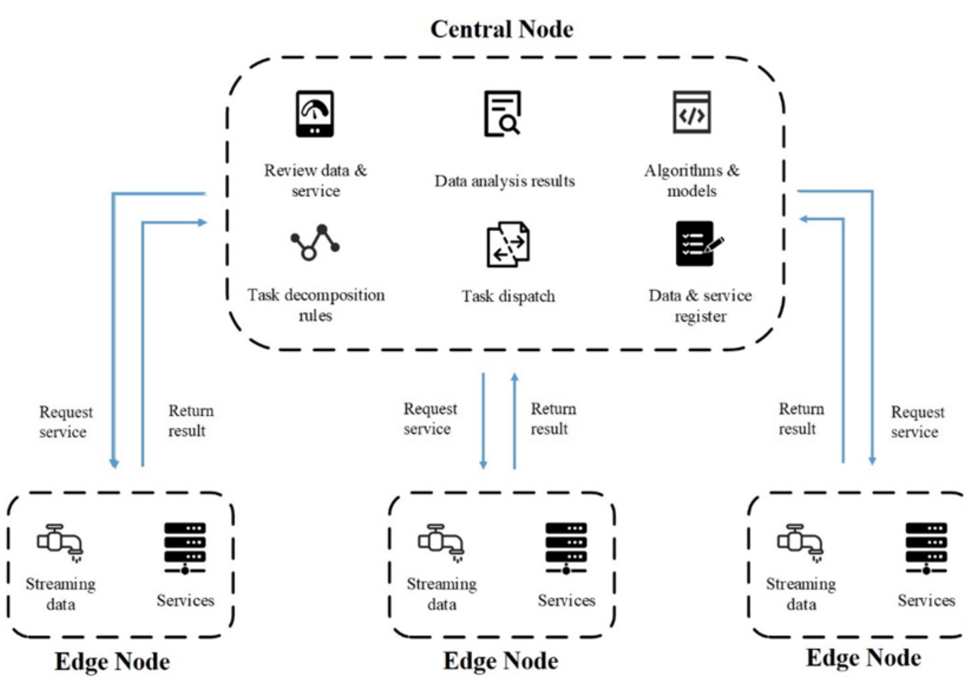

Fig. 2. Architecture of SCCS

ISSN: 2414-4592
DOI: 10.20474/jater-5.1.5

A. Preliminaries

In this section, we clarify the items used in the system, including application and service.

1) Application: Data analytics algorithm and model developed by domain-specified application developers and the essence of application is compute source code which represents the algorithm and model. Application can't be executed without data sources and computing resources.

2) Service: Application deployed in real computing environment which provides computing service to end users. Service is instanced by application specified data source and runtime environment.

B. System Users

SCCS has four types of users, application developers, administrator of central node, administrator of edge node and end user.

1) Application developer: Experts in a specific domain, who develop algorithms and models for analyzing the data in the system.

2) Administrator of central node: Responsible for the management of central node while reviewing data and services registered in the central node.

3) Administrator of edge node: Responsible for the management of central node while registering data to central node and deploying services in edge node.

4) End user: The actual user of services provided by system who analyses data in the system for a certain purpose.

C. System Nodes

SCCS consists of one central node and several edge nodes. Central node provides APIs for data and service register and supervises the status of data and services. Edge node receives and stores the streaming data generated by external business system. For a new service request, central node instantiates the abstract service according to the service registry information and dispatches the computing tasks to some edge nodes. Edge nodes conduct the tasks and return the mid-result to central node where the result is generated.

1) Central node: Central node is at the center of the network topology and monitors data and services in system. Central node schedules streaming data analyzing tasks collectively over edge nodes. Central node neither conducts data analysis task nor stores or generates streaming data. Concretely, central node is responsible for:

Register data and service: Only administrator of edge node register streaming data in edge node to central node, can the data be visible by application developers and end users. Applications for analyzing data can be developed for one or more kind of data and administrator of edge node can deploy and register the applications as services on edge node, where is near the streaming data source.

Interactive with users: End users can issue a streaming data analysis task through central node. Based on the register information, the task is decomposed and dispatched to edge nodes, which process is explained later. Data analysis results are returned to users through central node.

2) Edge node: Edge node is at the edge of network topology where the streaming data is generated or stored and the services are deployed. Edge node provides storage and computing resources while conducts the majority analysis tasks. Streaming data is analyzed by requesting services instead of sharing the raw data, which promises the data privacy and data value. Functionally edge node can:

Provide streaming data: streaming data is generated or received at edge node. Through Nginx load balance server, streaming is stored in Kafka, a high-throughput distributed streaming platform. When streaming data is needed, data consumer get streaming data through Kafka.

Provide service: edge node downloads data analyzing application from central node and specifies data sources and computing resources for the application, then application can provide data analyzing service to end users.

D. Service Composition

Based on our previous work [23], services can be divided into two types in our system: entity services and abstract services. Entity services are executable, indivisible atomic services running on edge nodes. Abstract services are the services that are not directly executable. Abstract services can be instantiated as a combination of one or more entity services, which has the same data process logic.

Relationships between services include instantiation relation and composition relation:

Instantiation relation indicates that one abstract service consists of one or more entity service.

Abstract service := entity service entity service

Composition relation indicates that one abstract service consists of other abstract services.

Abstract service := abstract service abstract service

Inspired by the service presentation method, data can be described as entity data and abstract data. Entity data represents the raw data stored on edge nodes, which can be analyzed directly. Abstract data represents one kind of data with common meanings, which need to be instantiated before computing.

When services are instantiated, the instantiation of the data are completed. In the other word, an abstract service is in-

ISSN: 2414-4592
DOI: 10.20474/jater-5.1.5

stantiated as serval entity services while every entity service is assigned to a data source. In our system, one abstract service can be instantiated serval sets of entity services, which means the task represented by this abstract service have different solutions.

IV. CASE STUDY

In this section, we use real shared bicycle data to verify the availability of streaming edge computing in our system. End user needs to analyze the distribution of shared bicycles in Beijing in order to adjust the number of shared bicycles in different areas to meet customers' demand. In Beijing even most cities in China, the bike-sharing business is operated and managed independently by several different companies. Large amounts of streaming business data generated in real time, which contains transaction records, user information and other private data, make it infeasible to concentrate all streaming data on cloud servers to analyze. In our system, the business system of each company acts as edge node providing the service analyzing the distribution of own shared bicycles. Central node requests services from each company and analyzes the final distribution of shared bicycles. On the one hand, majority computing tasks are conducted on edge nodes reducing the workload on central node and the network bandwidth for transmission. On the other hand, raw data is not transmitted.

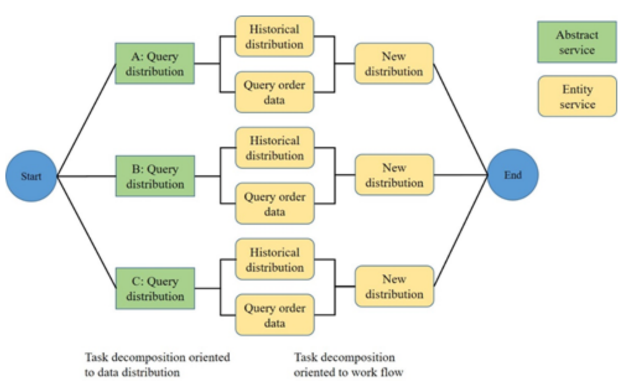

Fig. 3. Task decomposition oriented to data distribution and work flow

The process of analyzing the distribution of shared bicycles is an abstract service in our system, which can be decomposed into 3 abstract services provided by edge node A, B and C. Each abstract service can be instantiated as 3 entity services: query the historical distribution, query new order data and analyze the new distribution. And the process is shown in Figure 3.


```
{
    "ID": "634858431070208874",
    "COMPANY": 110001,
    "VEHICLE": 12229224,
    "LOCK": "1KB171713144",
    "ORDERTIME": 20180324233011,
    "VEHICLETIME": 20180324233011,
    "BACKTIME": 20180325000001,
    "STARTLON": 116.30797,
    "STARTLAT": 39.856697,
    "ENDLON": 116.287674,
    "ENDLAT": 39.85625,
    "COST": null,
    "MILES": 1.7,
    "STATUS": false,
    "BIRTH": 197903,
    "GENDER": false,
    "UNKNOW1": 116.314053,
    "UNKNOW2": 39.857968,
    "UNKNOW3": 116.29372,
    "UNKNOW4": 39.857489
}
```


Fig. 4. A example of real shared bicycle order data

A piece of real shared bicycle order data is shown in Fig xx. Due to the business data like ID, COMPANY, COST and customer privacy like BRITH, GENDED, the raw data is not suitable for network transmission. In SCCS, the raw data is processed on edge node first and an intermediate outcome that does not involve privacy is upload to central node. Because of the process of edge computing, the lower network bandwidth is needed and more secure the data privacy. Fig xx shows the final outcome of the distribution.

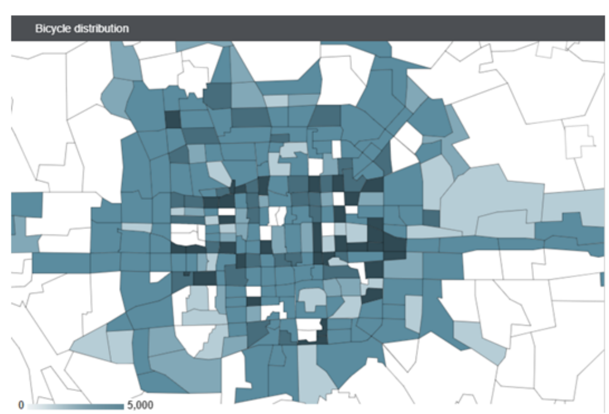

Fig. 5. The result of shared bicycle distribution

ISSN: 2414-4592
DOI: 10.20474/jater-5.1.5

V. CONCLUSION

In this paper, we present SCCS, a streaming cooperative computing system for edge environment. Our system can schedule edge nodes processing streaming data collectively. Users can define own data analyzing models and algorithms to meet their requirements regardless of the distribution of streaming data. The case study of shared bicycles indicates this system can coordinate multiple edge nodes for streaming data processing when the data is geographically distributed. In future work, we will improve the cooperative computing logic over edge nodes expecting complex data processing logic such as machine learning can be handled. For a start, we have finish the work that edge nodes can performance the data pre-processing task now. Meanwhile, we will develop graphical interface for developers without domain expertise.

ACKNOWLEDGEMENT

This research was supported by the National Key R&D program (No. 2016YFC0801700), Beijing Municipal Science and Technology Project (No. Z171100000917016), Beijing Municipal Transportation Commission (No. Z171100005117001).

REFERENCES

[1] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, ``Fog computing: Focusing on mobile users at the edge,'' 2015. [Online]. Available: https://bit.ly/2OctR7z

[2] C. H. Huang, M. H. Chang, and I. H. Lin, ``Research on single-board computers clustering the computing performance,'' *International Journal of Technology and Engineering Studies*, vol. 2, no. 5, pp. 125-133, 2016. doi: https://doi.org/10.20469/ijtes.2.40001-5

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, ``Edge computing: Vision and challenges,'' *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016. doi: https://doi.org/10.1109/jiot.2016.2579198

[4] E. B. Nejad and R. A. Poorsabzevari, ``A new method of winner determination for economic resource allocation in cloud computing systems,'' *Journal of Advances in Technology and Engineering Research*, vol. 2, no. 2, pp. 12-17, 2016. doi: https://doi.org/10.20474/-jater2.1.3

[5] B. Singh, S. Dhawan, A. Arora, and A. Patail, ``A view of cloud computing,'' *International Journal of Computers & Technology*, vol. 4, no. 2b1, pp. 387-392, 2013. doi: https://doi.org/10.24297/ijct.v4i2b1.3226

[6] S. Rodrigues, F. Faria, A. Ivaki, N. Cafofo, X. Chen, and F. Morgado-Dias, ``Tesla powerwall in the United States and Portugal: A comparative analysis on the use of storage with small scale photovoltaic systems,'' *International Journal of Technology and Engineering Studies*, vol. 2, no. 1, pp. 5-12, 2016. doi: https://doi.org/10.20469/ijtes.2.40002-1

[7] R. Roman, J. Lopez, and M. Mambo, ``Mobile edge computing: A survey and analysis of security threats and challenges,'' *Future Generation Computer Systems*, vol. 78, pp. 680-698, 2018.

[8] M. Dhiyanji and K. Sundaravadivu, ``Application of soft computing technique in the modelling and prediction of gold and silver rates,'' *Journal of Advances in Technology and Engineering Research*, vol. 2, no. 4, pp. 118-124, 2016. doi: https://doi.org/10.20474/jater-2.4.3

[9] H. Suo, Z. Liu, J. Wan, and K. Zhou, ``Security and privacy in mobile cloud computing,'' in *9th International Wireless Communications and Mobile Computing Conference (IWCMC),* New York, NY, 2013.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, ``Fog computing and its role in the internet of things,'' in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing,* New Dehli, India, 2012.

[11] I. Stojmenovic and S. Wen, ``The fog computing paradigm: Scenarios and security issues,'' in *2014 Federated Conference on Computer Science and Information Systems,* Tornoto, Canada, 2014.

[12] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, ``Mobile edge computing: A key technology towards 5G,'' *ETSI White Paper Journal*, vol. 11, no. 11, pp. 1-16, 2015.

[13] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, ``Mobile edge computing potential in making cities smarter,'' *IEEE Communications Magazine*, vol. 55, no. 3, pp. 45-50, 2017. doi: https://doi.org/10.1109/MCOM.2017.1600249CM

[14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, ``A survey on mobile edge computing: The communication perspective,'' *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, 2017. doi: https://doi.org/10.1109/COMST.2017.2745201

[15] C. Borcea, D. Iyer, P. Kang, A. Saxena, and L. Iftode, ``Cooperative computing for distributed embedded systems,'' in *Proceedings 22nd International Conference on Distributed Computing Systems,* California, CA, 2002.

[16] D. Ardagna and B. Pernici, ``Adaptive service composition in flexible processes,'' *IEEE Transactions on Software Engineering*, vol. 33, no. 6, pp. 369-384, 2007. doi: https://doi.org/10.1109/TSE.2007.1011

[17] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl, ``Qos aggregation for web service composition using workflow patterns,'' in *Proceedings of Eighth IEEE International Enterprise Distributed Object Computing Conference,* Beijing, China, 2004.

[18] M. Alrifai and T. Risse, ``Combining global optimization with local selection for efficient qos-aware service composition,'' in *Proceedings of the 18th International Conference on World Wide Web,* Bangkon, Thailand, 2009.

[19] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, ``Big data caching for networking: Moving from cloud to edge,'' *IEEE Communications Magazine*, vol. 54, no. 9, pp. 36-42, 2016. doi: https://doi.org/10.1109/MCOM.2016.756518501789330

[20] H. P. Sajjad, K. Danniswara, A. Al-Shishtawy, and V. Vlassov, ``Spanedge: Towards unifying stream processing over central and near-the-edge data centers,'' in *IEEE/ACM Symposium on Edge Computing (SEC),* Istanbul, Turkey, 2016.

[21] W. Yang, X. Liu, L. Zhang, and L. T. Yang, ``Big data real-time processing based on storm,'' in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications,* New Jersy, NJ, 2013.

[22] A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham *et al.*, ``Storm@ twitter,'' in *Proceedings of the ACM SIGMOD International Conference on Management of Data,* Dubai, UAE, 2014.

[23] Y. Zhang, H. Wang, J. Zhao, and B. An, ``SeCEE: Edge environment data sharing and processing framework with service composition,'' in *International Conference of Pioneering Computer Scientists, Engineers and Educators,* Singapore, 2018.