**Access Point Decoding Coded MapReduce for Tree Fog Network**

Xin Pang, Zhisong Bie, Xuehong Lin

School of Information and Communication Engineering,

Beijing University of Posts and Telecommunications, Beijing 100876, China

jiangnanmojian@gmail.com, zhisongbie@bupt.edu.cn, xhlin@bupt.edu.cn

**Abstract:** Recently, a Hybrid Coded MapReduce for server-rack architecture has been proposed. In the data shuffling phase, the total communication cost is divided into intra-rack communication cost and cross-rack communication cost. Compared with the general Coded MapReduce, intra-rack communication cost is reduced with increased cross-rack communication cost. We consider the three-tier tree topology under the Fog Computing environment, in which fog nodes are connected wirelessly through access points. The communication cost is divided into uplink and downlink cost, and computing and storage can be executed in access points. We propose Access Point Decoding Hybrid Coding MapReduce scheme under the Fog Computing environment. Theoretical analysis and simulation verification show that the total communication cost of secondary access points during the data shuffling phase can be reduced further compared with the Hybrid Coded MapReduce in the case of redundancy factor of 2.

**Keywords:** Fog Computing; Edge Computing; coding; MapReduce

**1 Introduction**

With the continuous development of the Internet of Things, more and more information terminals such as sensors, wearable devices, and home appliances are connected to the network, leading to explosive data growth. According to IERC, there will be 50 to 100 billion terminals accessing the network by 2020 [1]. Cisco predicts that by 2019 the amount of data generated by networked devices will reach 507.5 ZB per year [2]. The core network is faced with great pressure with massive heterogeneous data flooding into the network. Heterogeneous, low-latency, and dense network access and service cannot be satisfied by traditional centralized Cloud Computing architecture. The Fog Computing paradigm was first proposed by Cisco's Flavio Bonomi in 2011 [3]. In this paradigm computing, communication, control, storage resources and services are distributed to fog nodes close to users, solving the problems of Cloud Computing such as poor mobility, weak geographical information perception and high latency, and supporting more and more Application scenarios, including Internet of Things, 5G, embedded AI, etc.

MapReduce is a parallel programming model for large-scale data processing [4]. It is widely used in distributed search, distributed sorting, log analysis, machine learning, and other big data processing applications. Compared with the traditional parallel programming model, the MapReduce programming model has the advantages of transparent, fault-tolerant and extensible. The MapReduce execution process is mainly divided into map phase, shuffle phase, and reduce phase. In the map phase, input data blocks are assigned to different computing nodes, and each data block is mapped as an intermediate key-value pair as the input of the shuffle phase. During the shuffle phase, the intermediate key-value pairs are transferred among different nodes. In the reduce phase, existing intermediate key-value pairs in all nodes are summarized as the final results.

Reference [5] has pointed out that when using Hadoop clusters, 33% of the total task execution time is spent in the data shuffling phase. Code MapReduce was proposed in [6] to reduce the communication cost during data shuffling phase by redundant calculations and coded multicast. In [7] the application of the minimum bandwidth code, i.e., the Coded MapReduce described earlier and the minimum latency code combined with the MDS code in the Fog Computing environment was discussed. In [8], considering distributed matrix multiplication in the presence of straggling nodes, a unified framework combining Coded MapReduce with MDS code was proposed to reduces the impact from straggling nodes. Reference [9] has considered Coded MapReduce in wireless environment, and the downlink communication cost was reduced by random linear combination of code words. In [10] Hybrid Coded MapReduce for rack architecture was proposed to reduce cross-rack communication cost by centralizing cross-rack communications.

In this paper, we consider a three-tier tree wireless network topology in Fog Computing environment where the communication between fog nodes needs to go through certain access points. The communication cost is divided into the uplink communication cost and downlink communication cost of the access points, and computing and storage can be executed in access points. Since the rack architecture in [10] is actually a three-tier tree network topology and the Coded MapReduce is applicable for the Fog Computing environment, we propose a Access Point Decoding Hybrid Coded MapReduce scheme (ADHCMR) in which the access points participate in decoding under the Fog Computing environment. In the case of the redundancy factor of 2, compared with the Hybrid Coded MapReduce(HCMR) in [10], the total communication cost of secondary access points in the data shuffling phase can be reduced further. Theoretical analysis shows that the percentage decrease of our scheme is positively related to the system size, and it approaches a fixed value when the number of groups is very large. Simulation results show that the average percentage decrease is approximately 16.0% in the scenario where the number of groups is between 3 and 10.

**2 System model**

In this section, we introduce the system model of our scheme. Considering the three-tier tree wireless network topology of the Fog Computing architecture shown in Figure 1. Due to the limitation of the storage capacity of a single fog node (FN), multiple FNs have to cooperate to complete a given computing task. Before the map phase, each FN is assigned certain data blocks during off-peak hours. In the map phase, Intermediate key-value pairs are generated from the data blocks in FNs. Since each FN is responsible for reducing specific intermediate key-value pairs, intermediate key-value pairs of the same keys are sent to a specific FN during the shuffle phase. During the reduce phase, existing intermediate key-value pairs in each FN are summarized as the final output. We assume that the FNs within the coverage of the same secondary access point (AP) belong to the same group. The communication between the FNs in the same group only needs to go through the uplink and downlink of the local secondary AP, and the communication between FNs in different groups needs to go through the uplink and downlink of both the primary AP and several secondary APs. In the downlink phase, messages can be sent by the AP to multiple targets in the coverage at the same time in the form of wireless multicast.



**Figure 1** The three-tier tree wireless network topology of the Fog Computing architecture

Some parameters of the system are defined as follows: The redundancy factor of the map task, i.e., the number of FNs holding the same data block is denoted by *r*. The number of FNs in the entire system is denoted by *K*. The number of groups in the entire system is denoted by *P*, and *P* is equal to the number of secondary APs. The number of keys for the reduce phase is denoted by *Q*. The *ith* group is denoted by . The total number of input data blocks for a given computing task is denoted by *N*. The *jth* FN of the *ith* group in the system is denoted by. We assume that , , , . Each group contains the same number of FNs, and the same number of keys are reduced by each FN during the reduce phase.

**3 Proposed Access Point Decoding Hybrid Coded MapReduce scheme**

In this section, we will describe our ADHCMR scheme in detail and compare the uplink communication cost of the primary AP, denoted by , downlink communication cost of the primary AP, denoted by , uplink communication cost of secondary APs, denoted by , downlink communication cost of secondary APs, denoted by , and total communication cost of secondary APs, denoted by  during the data shuffling phase with HCMR based on the assumptions of two schemes from the perspective of theoretical analysis. As general Coded MapReduce, our scheme can be divided into map phase and shuffle phase.

**3.1 Map phase**

First, all nodes are divided intolayers, i.e., belong to the same layer, and we call it layer . Then *N* input data blocks are evenly assigned to  layers, and each layer containing  data blocks. The data blocks of the *jth* layer are evenly divided into  pieces, each corresponding to a single-layer FN set with a length of *r*. We assume that  is an integer multiple of  and the number of data blocks contained in each piece is *M*. For each data block of length *M*, it is assigned to all FNs in the corresponding set of length *r*, then the operations are repeated in other layers. The data block allocation phase ends.

Then the intermediate key pairs in each FN are generated from the assigned data blocks. If current FN and other FNs corresponding to certain keys belong to the same group (including the current node itself), all data blocks of current FN are used to generate intermediate key-value pairs corresponding to those keys above. Otherwise, since there is redundancy in the same layer, it is not necessary to use all the data blocks for such keys. Specifically, it is only necessary to satisfy the encoding and decoding of the subsequent data shuffling phase.

**3.2 Shuffle phase**

For a FN set *S* with a length of  in the layer  and a FN  in *S*, the data block set corresponding to  during the data block allocation phase is denoted by *D* with the length of *M*. denotes the set of intermediate key-value pairs needed by the group of  and generated from the data blocks corresponding to . We assume that the length of is an integer multiple of *r*. is divided evenly into *r* pieces, and each piece is in the one-to-one correspondence to a FN in . Then the operations above are repeated by other FNs in *S*. After that the set of intermediate key-value pairs corresponding to a single FN can be described by a two-dimensional matrix of *r* rows, where each row contains intermediate key-value pairs from a particular . We assume that the matrix has *x* columns, and each column can constitute a code word by xor. *x* code words are generated by each FN from its matrix and sent to local secondary AP through the secondary uplink. Then these code words are sent by the secondary AP to the primary AP through the primary uplink. Then these code words are multicast by the primary AP to the secondary APs corresponding to the groups that need these code words via primary downlink. After that all the operations above are repeated in other FN sets of length , then all the operations above are repeated in other layers.

Then for a certain FN  in the group , intermediate key-value pairs generated from the existing data blocks in  that can decode the existing code words of the secondary AP in  are sent to the secondary AP through the secondary uplink. Then the intermediate key-value pairs needed by the FNs in  are decoded by the secondary AP. Since there is no redundancy in the data blocks which belong to different FNs of one group, the intermediate key-value pairs generated from these data blocks can only be unicast. Then these intermediate key-value pairs generated from the existing data blocks in  and needed by other FNs in  are generated and sent to local secondary AP through the secondary uplink, then the operations above are repeated by other FNs in . After the previous steps, including the secondary AP, the intermediate key-value pairs distributed in  already satisfy the reduce phase of all FNs in . The existing intermediate key-value pairs of local secondary APs are sent through the secondary downlink to corresponding FNs. Then all the operations above are repeated in other groups. Finally, the intermediate key-value pairs owned by each FN already satisfy its reduce phase, and the shuffle phase ends.

**3.3 Analysis of HCMR**

 (1)

 (2)

 (3)

 (4)

*Proof.* During cross-group(i.e. cross-rack in [10]) communication phase there are  data blocks in each layer, which are divided into  pieces of size *M*. In coding phase, for each FN set of length  and each layer, the number of generated intermediate key-value pairs is  multiplied by the number of keys reduced by one group  and multiplied by . There are  layers and  FN sets of length  in each layer. Since each code word is sent through the uplink of the secondary AP close to sender, the uplink of the primary AP, the downlink of the primary AP and the downlink of the secondary APs in turn to *r* FNs. All communication cost of this phase includeand 

 (5)

and the first part of  denoted by 

 (6)

and the first part of  denoted by .

 (7)

At the beginning of the intra-group communication phase, for a single FN, the intermediate key-value pairs needed by the other FNs of the same group and generated by the layer which current FN belongs to are already concentrated on current FN. For these intermediate key-value pairs, the number of corresponding data blocks is equal to the number of single-layer data blocks , and the number of corresponding keys is equal to the sum of the number of keys needed by other FNs in the same group . Since there is no redundancy, these intermediate key-value pairs are not encoded and are sent to corresponding FN through the uplink and the downlink of local secondary AP. All communication cost of this phase include the second part of  denoted by  and the second part of  denoted by .

 (8)

We can get (1) to (4) by simplifying (5) to (8).

**3.4 Analysis of ADHCMR**

 (9)

 (10)

 (11)

 (12)

*Proof.* The cross-group communication phase is the same as HCMR except that code words are sent to the secondary APs instead of the FNs of the same layer, which excludes the downlink of the secondary APs. All communication cost of this phase include  and 

 (13)

and the first part of denoted by .

 (14)

Next, intermediate key-value pairs for decoding are sent by each FN to local secondary AP through the uplink of local secondary AP. After the intermediate key-value pairs needed by FNs in the same group are decoded by the secondary APs, these intermediate key-value pairs decoded are sent to corresponding FNs by the secondary APs through their downlink. Because each code word in the cross-group communication phase is sent to *r* different secondary APs, the number of code words owned by all secondary APs is , each composed of *r* intermediate key-value pairs.  intermediate key-value pairs are used to solve 1 intermediate key-value pair. All communication cost of this phase include the second part of denoted by 

 (15)

and the first part of denoted by .

 (16)

Finally, for those intermediate key-value pairs for intra-group communication, the number of data blocks owned by each FN is , and the number of corresponding keys is equal to the sum of the number of keys needed by other FNs in the same group. These intermediate key-value pairs are sent through the uplink and the downlink of local secondary AP. All communication cost of this phase include the third part of denoted by and the second part of denoted by .

 (17)

We can get (9) to (12) by simplifying (13) to (17). From the previous analysis, the total communication cost of primary AP of ADHMCR and HCMR are the same. In this paper, we care more about the total communication cost of the secondary APs. Then we compareand . Let and associate (4) and (12), after simplification we can get the inequality.

 (18)

The solution of this inequality is

 (19)

We can see that whenand, , and when and, . It is intuitively understood that the number of decoding intermediate key-value pairs sent to local secondary APs via the uplink of the secondary APs is positively correlated with *r*. Compared with the HCMR where FNs in the same layer decode, with *r* increases in ADHCMR, the uplink communication cost of the secondary APs will increase. We set  as the decrease of the total communication cost of the secondary APs during the data shuffling phase of our scheme, and we can get

 (20)

then  according to our scheme, so

 (21)

It is easy to see that when , i.e., the number of group is equal to the number of layer(we call this *equal state*), reaches the maximum value .

 (22)

The percentage decrease of  is denoted as , and we can get

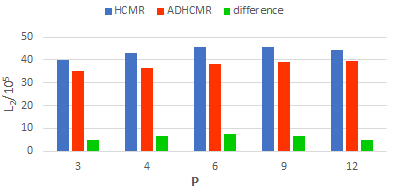
 (23)

We can see that in *equal state*,  increases as *P* increases, and when *P* is very large, infinitely approaches 28.6%.

**4 Simulations and results**

In this section, we present the simulations and results of HCMR and our ADHCMR. The parameters are picked to satisfy the theoretical settings of two schemes.

In Figure 2 we assume that , , ,  and we vary  for two schemes. The total communication cost of the secondary APs in HCMR is denoted by blue, and the total communication cost of the secondary APs of our ADHCMR is denoted by red, and the decrease of our scheme is denoted by green. Note that the ordinate value is denoted by of the actual cost, we can observe that when , i.e., the *equal state*. The largest decrease is achieved and the average percentage decrease in Figure 2 is approximately 13.9%.



**Figure 2** The influence on total communication cost of the secondary APs during the data shuffling phase from the relationship between *P* and *K*

In Table I we assume that , , ,  and we vary . The decrease of our scheme is denoted by, and the percentage decrease corresponding to  is denoted by . *N* is picked as the lower limit of two schemes for corresponding parameters. Considering the better performance in *equal state*  is picked. As shown in Table I, we can note that the lower cost is achieved by our scheme compared with HCMR. The  of two schemes,  and  grow as the system scale grows. With 3 groups, there is only 6.1% decrease, and with 10 groups, there is already 21.3% decrease.  gets more and more closer to the theoretical upper limit. The average percentage decrease in Table I is approximately 16.0%.

**Table** **I** Comparison of the total communication cost of the secondary APs during the data shuffling phase of two schemes as the system scale increases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| (9,3,9,18,2) | 297 | 279 | 18 | 6.1% |
| (16,4,16,48,2) | 1728 | 1536 | 192 | 11.1% |
| (25,5,25,100,2) | 6250 | 5350 | 900 | 14.4% |
| (36,6,36,180,2) | 17280 | 14400 | 2880 | 16.7% |
| (49,7,49,294,2) | 40131 | 32781 | 7350 | 18.3% |
| (64,8,64,448,2) | 82432 | 66304 | 16128 | 19.6% |
| (81,9,81,648,2) | 154548 | 122796 | 31752 | 20.5% |
| (100,10,100,900,2) | 270000 | 212400 | 57600 | 21.3% |

**5 Conclusions**

In this paper, we proposed ADHCMR for a three-tier tree wireless network topology in Fog Computing environment to reduce further the total communication cost of the secondary APs during the data shuffling phase when redundancy factor is equal to 2. Compared with HCMR, in the *equal state*, the percentage decrease grows as the system scale grows and approaches its theoretical upper limit.

**References**

1. H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, Vision and challenges for realising the Internet of Things, Cluster of European Research Projects on the Internet of Things—CERP IoT, (2010).
2. C. V. Networking, Cisco global cloud index: Forecast and methodology 2014-2019 (white paper)[R]. 2013.
3. Bonomi F, Milito R, Zhu J, et al. Fog computing and its role in the internet of things[C]// Edition of the Mcc Workshop on Mobile Cloud Computing. ACM, 2012:13-16.
4. Dean J, Ghemawat S. Map Reduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.
5. Chowdhury M, Zaharia M, Ma J, et al. Managing data transfers in computer clusters with orchestra[J]. Acm Sigcomm Computer Communication Review, 2011, 41(4):98-109.
6. S. Li, M. A. Maddah-Ali and A. S. Avestimehr, "Coded MapReduce," 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, 2015, pp. 964-971.
7. S. Li, M. A. Maddah-Ali and A. S. Avestimehr, "Coding for Distributed Fog Computing," in IEEE Communications Magazine, vol. 55, no. 4, pp. 34-40, April 2017.
8. S. Li, M. A. Maddah-Ali and A. S. Avestimehr, "A Unified Coding Framework for Distributed Computing with Straggling Servers," 2016 IEEE Globecom Workshops (GC Wkshps), Washington, DC, 2016, pp. 1-6.
9. S. Li, Q. Yu, M. A. Maddah-Ali and A. S. Avestimehr, "A Scalable Framework for Wireless Distributed Computing," in IEEE/ACM Transactions on Networking, vol. 25, no. 5, pp. 2643-2654, Oct. 2017.
10. S. Gupta and V. Lalitha, "Locality-aware hybrid coded MapReduce for server-rack architecture," 2017 IEEE Information Theory Workshop (ITW), Kaohsiung, 2017, pp. 459-463.