Research and Realization of AP Clustering Algorithm Based on Cloud Computing

Yue Qiang[1, a *], Hu Zhongyu[2, b], Lei Xinhua[1, c], Li Xiaoming[3, d]

[1] School of Information Technology, Kunming University, Kunming 650214, China

[2] School of Auto-control and Mechanical Engineering, Kunming University, Kunming 650214, China

[3] Office of Science Research, Kunming University, Kunming 650214, China

[a]wallay@126.com, [b]poundblue@126.com, [c]lkxmh@sina.com, [d]lxm@sina.com

Keywords: cloud computing; Hadoop; MapReduce; AP clustering algorithm; community structure

Abstract. With the extensive application of network, massive growth in the scale of data through cloud computing has been observed. Cloud computing is a powerful technology to perform complex computing, and applications running on cloud computing with Hadoop architecture are increasing. In this paper, after studying the AP clustering algorithm, we proposed the AP clustering algorithm based on MapReduce model and realized the parallelizing AP clustering algorithm in the cloud computing platform of Hadoop. We tested the parallelizing AP clustering algorithm by using two real-word networks. The experimental results show the capability of our algorithm successfully to detect community structure in complex networks.

1. Introduction

Cloud computing is a significant technology to perform massive-scale complex computing and has become a powerful architecture. It can significantly reduce the cost of hardware, service, and software [1]. To analyze the huge amounts of data for extracting meaningful information, there is requirement of deploying data intensive application. The advantages of cloud computing include parallel processing, virtualized resources, security, and data service integration with scalable data storage. Cloud computing can not only minimize the cost for automation and computerization of individuals and enterprises, but also can reduce the cost of infrastructure maintenance, efficient management, and user access.

Hadoop is an open-source Apache software project based on java that enables the distributed processing of large datasets across clusters, and it is a reliable and scalable software framework for parallel and distributed computing. Hadoop has two primary components, HDFS and MapReduce. HDFS is a distributed file system designed to run on top of the local file systems of the cluster nodes and store extremely large files suitable for streaming data access. MapReduce is a simplified programming model for processing large amount of datasets pioneered by Google for data-intensive applications. Therefore, the storage system is not physically separated from the processing system. Traditional and Existing tools and applications become deficient to process large amount of data. Hadoop has the ability to solve the problem of handling and processing large-scale data that terabytes and petabytes sized. Many enterprises, companies and universities deploy Hadoop clusters in highly scalable and elastic computing environments.

The rest of this paper is organized as follows. Section 2 introduces the cloud computing with Hadoop architecture. Section 3 introduces the MapReduce programing model. Section 4 describes the AP clustering algorithm. Section 5 realizes the AP clustering algorithm based on MapReduce. Section 6 presents and analyzes the experimental results. Section 7 concludes this paper.

2. Cloud computing with Hadoop architecture

Hadoop is a preferential choice in open source cloud computing community for providing an efficient platform for Big Data, and it provides an extensive selection of effective cloud to help users achieving their goals [2]. Hadoop is not only a distributed file system with storage function, and also

a framework to perform distributed application on the large clusters consisting of general-purpose computing devices. Hadoop consists of the following projects, as shown in Fig. 1.

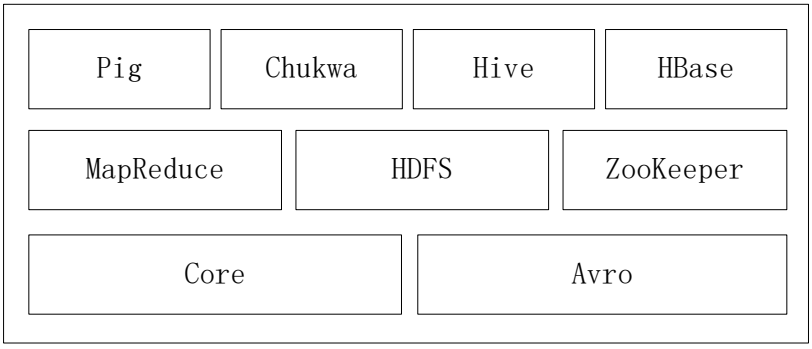

Fig. 1 Hadoop architecture

Pig involves a high-level scripting language and offers a run-time platform that allows users to execute MapReduce on Hadoop. Hive offers a warehouse structure in HDFS. Hbase offers a scalable distributed database that supports structured data storage for large tables. Chukwa is a data collection and analysis framework incorporated with HDFS. Zookeeper is high-performance service to coordinate the processes of distributed applications. Avro is a data serialization system, and the tasks performed by Avro include data serialization, remote procedure calls, and data passing from one program or language to another. The most significant feature of Hadoop is that HDFS and MapReduce are closely related to each other.

3. MapReduce model

MapReduce is programming model or a software framework used in Apache Hadoop, and it can help a programmer with less experience to write parallel programs and create a program capable of using computers in a cloud. Hadoop MapReduce is provided for writing applications which process and analyze large data sets in parallel on large multi-node clusters of commodity hardware in a scalable, reliable and fault tolerant manner [3]. Data analysis and processing specifies two functions namely, the map function (mapper) and the reduce function (reducer). The mapper regards the key/ value pair as input and generates intermediate key/value pairs. The reducer merges all the pairs associated with the same key and then generates an output. The map function is applied to each input (key1, value1), where the input domain is different from the generated output pairs list (key2, value2), and the reduce function is applied to each [key2, list (value2)] to generate a final result list (key3, value3). MapReduce programming model is shown in Fig. 2.

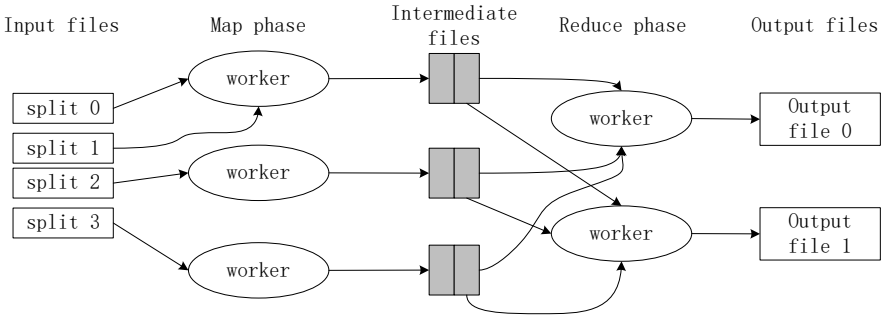

Fig. 2 MapReduce programming model

MapReduce uses master/slave architecture. In the architecture, JobTracker daemon runs on master node and TaskTracker daemon runs on each salve node. JobTracker and TaskTracker are known as the MapReduce engine [4].

(1) JobTracker. JobTracker is in charge of scheduling all jobs, and it is the core of the system assigning tasks. JobTracker runs on master node and monitors MapReduce tasks executed by TaskTracker on salve nodes. JobTracker is unique in master/slave architecture, and it is the only responsible for the control of MapReduce in a Hadoop system. The job is submitted to JobTracker on the Master node, and then JobTracker asks NameNode for the actual location of data in HDFS to be

processed. JobTracker locates TaskTracker on slave nodes and submits the jobs to TaskTracker on slave nodes.

(2) TaskTracker. A TaskTracker is in charge of executing user-defined operations. A TaskTracker runs on slave nodes. It accepts jobs from JobTracker and executes MapReduce operations. A TaskTracker sends the "heartbeat" message to JobTracker during executing operations, and report about the execution status of each task. The "heartbeat" message permits JobTracker to know how many tasks are available in TaskTracker on a slave node. It is the responsibility of TaskTracker to help JobTracker collecting the whole situation of job execution and providing important gist for the distribution of the next task.

4 AP clustering algorithm

Affinity propagation(AP) clustering algorithm is an high-efficient clustering method proposed by J.Frey in 2007 [5], and it has been shown to create clusters in much less time, and with much less error than traditional clustering algorithms (such as K-means). Unlike the k-means algorithm, which chooses a cluster center from subset of data points and need to specify the number of clusters, the AP algorithm considers simultaneously all data points as cluster centers and thus is independent of the quality oft he initial set of cluster centers. There are two types of messages that are passed between data points: responsibility and availability. The responsibility R(i,k) is sent from data point i to candidate exemplar point kand reflects how well suited it would be for point k to be the exemplar of point i.The availability A(i,k) is sent from data point candidate exemplarpoint k to data point i and reflects how appropriate it would be for data point i to choose candidate exemplar k as its exemplar. The similarity of each data points is set to a negative squared Euclidean distance between point i and j.

$$S(i,j) = -\sqrt{\sum_{k=1}^{n}\left(X_{ik} - X_{jk}\right)^2} \tag{1}$$

where X_{ik} and X_{jk} are the kth attribute of point X_i and point X_j.

The algorithm process is as follows:

(1) Initialization. The availabilities are initialized to zero.

(2) Calculate responsibilities according to Eq. (2).

$$\mathrm{R}(i,k) \leftarrow S(i,k) - \max_{k \neq k'}\{A(i,k') + S(i,k')\} \tag{2}$$

where $S(i,k)$ is the similarity between point i and k.

(3) Calculate availabilities according to Eqs. (3) and (4).

$$A(i,k) \leftarrow min\{0, \mathrm{R}(k,k)\} + \sum_{j \neq i,k} max\left(0, \mathrm{R}(j,k)\right) \tag{3}$$

$$A(k,k) \leftarrow \sum_{j \neq i,k} max\left(0, \mathrm{R}(j,k)\right) \tag{4}$$

(4) if the cluster centers do not chang or is greater than the maximum number of iterations
then return
else
jump to step (2).

5. Realization of AP clustering algorithm based on MapReduce

(1) Calculating the similarity matrix

Mapper gets the values of $(X_{ik}\text{-}X_{jk})^2$, then reducer adds the values and set to negative squared similarities. The key of input (key, value) is the number of row and column of input matrix, and the value is value of matrices of corresponding location.

(2) The distributed processing of AP clustering algorithm

Mapper converts each row of similarity matrix and availability matrix from input text to the format required with Eq. (2). Reducer merges similarity values and availability values in a same row after MapReduce shuffle phase, and calculates the responsibility values in a same row according to Eq. (2). The process of calculating the responsibility values is as Fig. 3.

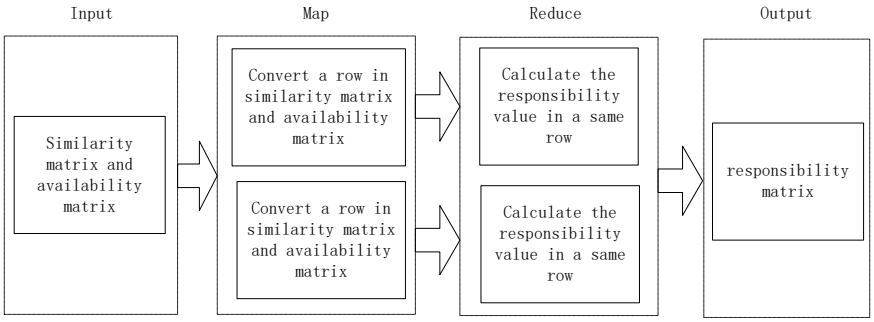

Fig. 4 The MapReduce processing of calculating responsibility matrix

(3) Determining the cluster centers

Cluster center is determined on responsibility R(k,k) and availability A(k,k) of data points k. When R(k,k)+ A(k,k)>0, that means the responsibility and availability of data points k are large enough for itself , and data point k is suitable for cluster center.

(4) Dividing the data points

After cluster centers are determined, all data points are needed to divide to different cluster. Data point i belongs to which cluster center is based on Eq. (5):

$$\max_{1\le k\le n}\{A(i,C_k)+R(i,C_k)\} \tag{5}$$

where n is the number of total cluster centers. C_k is the kth cluster center.

6. Experimental results and analysis

6.1. Experiment settings

Community structure detection in complex networks has been intensively investigated in recent years [6]. We use three interconnection computers to construct cloud computing environment. It is a master/slaver structure, and one computer is master and the other two are slavers. The cloud computing environment is shown in Table1.

Table 1 Configuration of cloud computing environment

Host	IP Address	Hardware Configuration	Software Configuration
Mater	192.168.1.1	CPU: Intel Xeon E5405 Memory: 4 GB	OS: Red Hat Linux 9.0 Hadoop 0.20.203.0
Slaver1	192.168.1.2	CPU: Intel Core2 Q9400 Memory: 2 GB	OS: Red Hat Linux 9.0 Hadoop 0.20.203.0
Slaver2	192.168.1.3	CPU: Intel Core2 Q9400 Memory: 2 GB	OS: Red Hat Linux 9.0 Hadoop 0.20.203.0

6.2. Performance evaluation

To evaluate the performance of AP clustering algorithm, EQ function is used as the criteria.

$$\mathrm{EQ}=\frac{1}{2m}\sum_i\sum_{v\in C_i,u\in C_i}O_vO_u\left(A_{vu}-\frac{k_vk_u}{2m}\right) \tag{6}$$

where m is the number of edges. O_v and O_u are the numbers of communities who including node v and node u. A is the adjacency matrix of network. k_v and k_u represent respectively the degrees of node v and node u. The higher EQ value means better overlapping community structure

6.3. Experimental results and analysis

To test AP clustering algorithm, we have performed our experiments over two real-world networks. We applied it to two widely used real-world networks with a known community structure. They are the well- known Karate Club network and Dolphin network. The Karate Club network is a network of friendship which has 34 members of a karate club as nodes and 78 edges representing friendship between members. Due to a leadership issue, the club splits into two distinct groups. The

Dolphin network is a community of dolphins living in New Zealand. There are 62 dolphins and edges are set between network members that are seen together more often than expected by chance. The network is split naturally into two large groups, and the number of edges is 159.

We compare AP clustering (**APC**) algorithm with the COPRA algorithm using run time and EQ to evaluate the performance. The results are compared with the COPRA algorithm and shown in Table 2.

Table 2 Comparison of performance measures

Algorithms	**Karate network**		**Dolphin network**	
	run time (ms)	EQ	run time (ms)	EQ
COPRA	164	0.158	224	0.303
APC	32	0.168	92	0.306

The APC algorithm outperforms the COPRA algorithm in our experiments with shorter run time and higher EQ values. Especially, when the network scale become huge, the APC algorithm can accomplish clustering task spending shorter time than COPRA algorithm. Therefore, the distributed APC algorithm can deal with large-scale data within the limited time.

7. Conclusions

In this paper, we have expounded the cloud computing with Hadoop architecture and the work mechanism of the MapReduce model. MapReduce is the most popular distributed computing framework. Then we have researched AP clustering algorithm and realized it on MapReduce. Compared with other traditional clustering algorithms (ex. K-Means), APC algorithm does not need to specify the number of clusters and choose a cluster center from subset of data points. We constructed cloud computing environment to realize parallelizing APC algorithm for identifying a community in complex networks. The performance of our algorithm was tested on two real-world networks. Experimental results confirm the validity and advantage of this approach. As future work, we will aim at combining our method with other methods to improve the quality of results.

Acknowledgement

In this paper, the research was sponsored by the Science Research Project of Kunming University (Project No. XJL15013) and Science Research Project of Kunming University (Project No. XJL14006).

References

[1] L. Chang, R. Ranjan, Z. Xuyun, Y. Chi, D. Georgakopoulos, C. Jinjun, Public Auditing for Big Data Storage in Cloud Computing – a Survey, Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on, 2013, pp. 1128–1135.

[2] P.D. Londhe, S.S. Kumbhar, R.S. Sul, and A.J. Khadse, “Processing big data using hadoop framework”, in Proceedings of 4th SARC-IRF International Conference, New Delhi, India, Apr. 27, 2014, pp. 72-75.

[3] J. Ekanayake, S. Pallickara, and G. Fox, “Mapreduce for data intensive scientific analyses”, in IEEE 4th International Conference on eScience, Indianapolis, Indiana, USA, Dec. 7-12, 2008, pp. 277-284.

[4] J. Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, Communications of the ACM 51(1).

[5] Frey B J, Dueck D. Clustering by passing messages between data points [J]. Science, 2007, 315(5814): 972-976.

[6] S. Sadi, Ş. Öğüdücü, A.Ş. Uyar, An efficient community detection method using parallel clique-finding ants, in: The 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain, 2010, pp. 1–7.