Parallel Design of Hash and K-means Algorithm in the Context of Big Data

Xing Lei, Zhang Xiang, Guo Zhengkun, Guo Fuwang

China Aviation Integrated Technology Research Institute, Beijing

queque0506@163.com

Keywords: K-means; Hash; mass data

Abstract: In order to further improve the efficiency of K - means algorithm on the large-scale data clustering, this paper conducts deep analysis and research on the optimization of K - means clustering algorithm and proposes a selected program of initial clustering center based on Hash algorithm, hashing mass high-dimensional data to a compression space to excavate the clustering relations, so as to make the selected initial clustering center tend to be convergent state as far as possible and to greatly reduce the number of iterations of clustering, improved the accuracy of clustering.

1. Introduction

The wave of Internet has spawned a large amount of data, which contains inestimable business value and guiding value. How to dig out useful information from these massive and disorderly data has become a quite important research topic [1]. In order to rapidly improve the efficiency of clustering algorithm in processing massive data sets, cluster resources can be used to efficiently perform mining tasks, and the improved clustering algorithm based on distributed computing platform Hash can effectively solve such problems. This paper studies the parallel implementation of K-means algorithm and its optimization algorithm on the Hash platform. On the one hand, due to the uncertainty of initial k value and the instability of random selection of initial cluster center of the K-means algorithm, this study proposes to initialize the k value and initial cluster center of the k-means algorithm based on the pre-clustering algorithm Canopy to improve the convergence speed of the algorithm and the stability of the clustering results. On the other hand, in order to make full use of the advantages of Hash characteristics, system configurations such as memory optimization, data compression, data serialization, running memory ratio, running heap ratio and cache size can be optimized to accelerate the improvement of the parallel computing efficiency of Canopy_K-means (CKM) algorithm and its application ability in the distributed computing environment.

2. K-means algorithm and Hash algorithm

2.1. K - means algorithm

The k-means clustering algorithm is based on division. The basic idea is as follows: the user randomly selects k points in the sample as the initial clustering center, classifies the points closest to each initial center as the class where the point is, and updates the location of the clustering center successively through iteration until the convergence of standard measure function [2]. The implementation steps are as follows:

step1 randomly select k clustering centers;
step2 classify each sample point into the center point nearest to it;
step3 reset the center points of each class;
step4 repeat step2 and step3 until convergence.

2.2. Hash algorithm

Hash algorithm, also known as Hashing algorithm, transforms the input of arbitrary length into

  DOI: 10.25236/mmmce.2019.129

the output of fixed length through the Hash function, and the output is the Hash value [3]. This transformation is a compression mapping in which the space of the hash value is usually smaller than the space of the input, and different inputs may be hashed into the same output. A properly designed Hash function is the key to the Hash algorithm. Obviously, the Hash function will map multiple input values to the same address space, which is called “conflict”, and the input values that collide are called “synonyms” [4].

2.3. Initial clustering center selection based on Hash algorithm

Algorithm is the basic idea of rational design of hash function similarity of large data hash to the same address space, different similarity data hash to a different address space, and then select k initial clustering center in k address space where the synonyms are the most; the initial clustering center selected can meet the principle of minimum distance within clusters and the maximum distance among the clusters[5]. The basis of this algorithm can be summarized as follows:

(1) massive and high-dimensional data are mapped to a compressed address space through the hash function, which facilitates the control of data clustering relations.

(2) after mapping, data with high similarity will “conflict” and become synonyms. The center of the densest data band is closest to the clustering center.

(3) by selecting different synonyms, the selected initial clustering center is not in the same class as far as possible.

(4) it avoids selecting the isolated point as the initial clustering center.

Definition 1 Step function is a linear combination of finite interval index functions. Step function y = f(x) is defined; the domain is [a, b]; and the range is {y1,y2...,yn}; interval length is d; then the step function can be expressed as:

$$f(x) = \begin{cases} y_1, [a,\ a+d) \\ y_2, [a+d,\ \&a+2d) \\ \cdots\cdots \\ y_k,\ [b-d, b] \end{cases} \quad (1)$$

Definition 2 Hash hash function is designed as:

$$H(key) = f(key) \quad (2)$$

Set the number of data set as n, attribute dimension as m; keyi(0< I <m) represents the ith dimension attribute of data; the maximum value is maxi; the minimum value is mini; the interval length is di = |maxi-mini|/k; the definition domain of function is [mini, maxi]; and the range is {yi1,yi2...,yik}; k represents the number of clusters; the function image is shown in Figure 1; and the hash relation is shown in Figure 2.

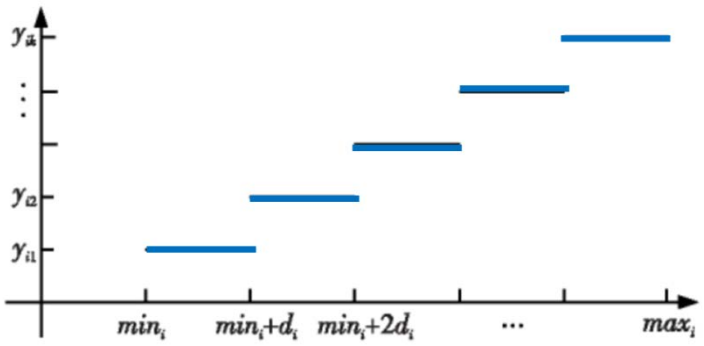

Figure 1. Hash function diagram

By definition 2, the hash function will hash the input data attributes of each dimension value to k address space; each dimension attribute of the data corresponds to a single address space; number each address space, and each data can use the row vectors made up of address space number;

transform them to matrix and make statistics of the number; find out the most k row vector, calculate the center of all the data corresponding to each row vector; it is concluded that the k center is the selected initial clustering centers.

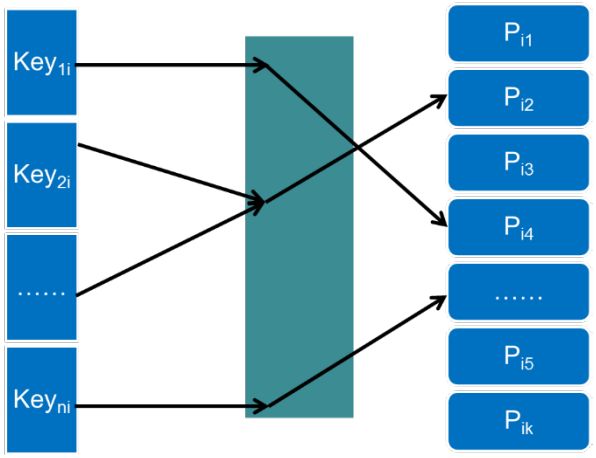

Figure 2. Hash diagram

3. Parallel of Hash and k-means algorithms in rongrong

To form the Internet platform of high-end equipment manufacturing industry, to connect enterprises, and to provide the basic, general, and comprehensive services as well as advisory services on personnel training and enterprise certification for the enterprises, rongrong constructs most powerful enterprise information integration service platform of high-end equipment manufacturing; it connects the high-end equipment manufacturing industry through information integration, constructs the largest enterprise operations management consulting service technology platform. The parallel algorithms of Hash and k-means applied in rongrong are introduced below.

3.1. Algorithm description

Input: dataset; k (number of clusters).
For i =1 to do//n represents the number of data
{
For j=1 to do//m represents the number of data
{
Max j=Get Max( )；
Min j= Get Min( )；
d＝ (Max j- Min j) / k；
if (Min jxj＜Minj＋kd) return Pj1；
if (Min j+dxj＜Minj＋2d) return Pj2；
// depends on the value of k
if (Min j+(k-1)d＜xjMinj＋kd) return Pjk；
} // each one-dimensional attribute corresponds to an address number
} // each data is converted into an address number string
The data with the same number string are saved together to form a data cluster;
Find the data cluster with the most k data;
The clustering center of these data clusters is calculated and used as the initial clustering center of k-meas algorithm.
Output: clustering results

3.2. Concrete implementation of algorithm parallelization

The implementation details of the algorithm are shown in Figure 3. Each job is different in

design according to its own tasks, and the key details are described as follows:

Job1:

(1) ket, formed after the data is split, is the byte offset of the current data relative to the starting position; value is the string composed of attributes of each dimension of the current data.

(2) The operation of Mapper is to analyze the value; compare the size of corresponding dimensions of data, save the maximum and minimum values of each dimension attribute, and output the number of each dimension attribute as a new key. Value saves the maximum and minimum values of this dimension.

(3) What the reducer operate is to parse the value, divide the attributes of each dimension into the corresponding address space, and number each address space.

(4) Output results; key is the number of the address space, and value is the value range of the address space.

Job2:

(1) Update the address space divided by Job 1 to Job2.

(2) The key, formed after the data is split, is the byte offset of the current data relative to the starting position, and value is the string composed of attributes of each dimension of the current data.

(3) The operation performed by Mapper is to parse the value, match the attributes of each dimension of data with the corresponding address space, record the corresponding address space numbers; each number corresponds to a dimension; each data can be represented by a number string, and the number string can be taken as a new key output.

(4) The operation performed by the Partitioner is the partition of data so that the data with the same number string is in the same partition.

(5) Combiner performs the operation of combining data of the same number and counting the number of such data.

(6) The key formed after data merge is the number string of data, and value is the string array composed of all the attributes of each dimension with the number string data.

(7) The operation performed by Reducer is to analyze the values of the first k arrays, calculate the central point of each array, and take these k central points as the initial clustering center of the k-means algorithm.

Job3:

(1) Update the central point calculated by Job2 to Job3.

(2) The key, formed after the data is split, is the value of the initial cluster center to which the current data belongs; value is the string composed of attributes of each dimension of the current data.

(3) The operation performed by Mapper is to parse the value, calculate the distance from each point to the clustering center, and classify the data into the classes where the nearest clustering center is located.

(4) The Partitioner aims to partition the data according to the cluster center so that the data belonging to the same class are in the same partition.

(5) Combiner performs the operation of parsing the value and adding the corresponding attributes of each dimension of each data. The new value consists of the accumulated values of the attributes of each dimension and the accumulated total number of data.

(6) The key formed after data merge is the value of the initial cluster center to which the current data belongs; value is an array composed of the values of all data belonging to the initial cluster center.

(7) What the Reducer operates is to parse all values in the array, calculate the new clustering center, and save it.

(8) Judge whether the termination condition is to terminate the clustering.

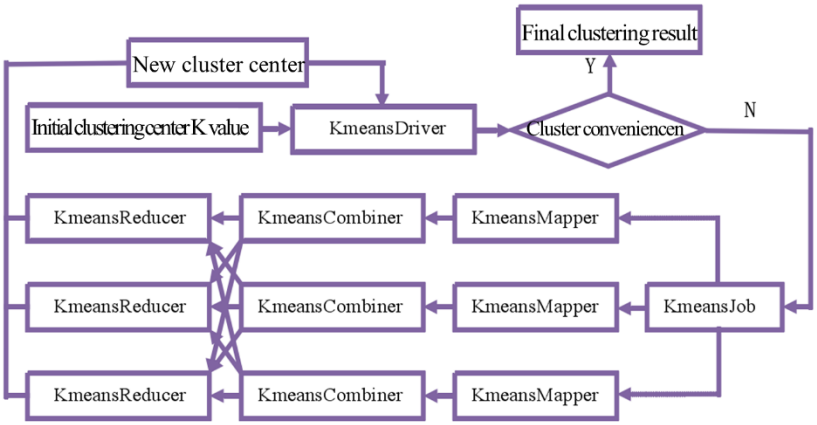

Figure 3 K-means clustering flow chart

References

[1] Liu Xingliang. Kmeans Analysis of Massive Book Circulation data Based on Hadoop [D]. East China Institute of Technology, 2015.

[2] Wang Pingxin, Shi, Hong, Yang, Xibei, et al. The Three - way k - means: integrating k - means and Three - way decision [J]. International Journal of Machine Learning and Cybernetics, 2019.

[3] Zhang Liwu, Feng Bao, Zhou Jianhua, et al. TCP Search Hash Algorithm for High Performance Computing Network [J]. Computer Technology and Development, 2018, 28(5):100-104.

[4] Lu Xinyu. Application of Consistent hash Algorithm in Logistics Data Platform [J]. Electronic World, 2019, 561(3):19-19.

[5] Li Wenfeng, Shen Lianfeng, Hu Jing. Optimization Algorithm of Adaptive Energy-saving Routing for Inter-cluster Communication in Sensor Networks [J]. Journal of Communications, 2012(3):10-19.