International Journal of Advanced and Innovative Research (2278-7844) / # 1/
Volume 6 Issue 11

Efficient Algorithm for Big Data Application

Santhiya R, Revathi M, Madanachitran R

Assistant Professor, Department of Computer Science and Engineering, Paavai Engineering College, Namakkal

ABSTRACT:
Data mining applications play an important role in IT firms where energy wastage is the main problem. Increase in workload and computation leads to high energy cost. Mapreduce scheduling algorithm is a model which is developed for processing and storing large volume of data at the same time. EMRSA is an algorithm gives reliable energy and reduction in maps based on arrangement priority based scheduling is provided to the test for utilization and system work is easily improved by reduction with maps.

Keywords: Big Data, EMRSA, Mapreduce, Incremental processing.

1 INTRODUCTION

Big data – both structured and unstructured – that overwhelms a business on a day-to-day basis. It's what organizations do with the data that matters. Big data can be analysed for visions that lead to well decisions and strategic business moves. The major areas covered finance, banking, education, E-commerce and so on.

Map reduce program is collected of map procedure that performs a summary operation. It is used to gather data according to the request. To progression big data proper scheduling is required to attain greater performance. Scheduling is a procedure of assigning jobs to available resources in a manner to diminish starvation and maximize resource utilization.

There is a tendency to focus on reduce. Currently implemented in bigdata application. Large data is constantly evolving. With the arrival of communication means such as new technologies, devices and social networking sites, the amount of data produced by humans is hastily increasing every year. All this data is meaningful and useful when treated, but it is ignored. Data essentially means large data.

It is a large data set that cannot be processed using traditional computing technology. Big data is not just large data; it is a complete subject counting a variety of tools, techniques and frameworks. As new data and updates are gathered, the input data of the large data mining algorithm changes increasingly and the result becomes out-of-date.

EMRSA METHODOLOGY:

In the current situation, energy waste is great. The problem is a lot of IT companies. More workload cunnings increase high energy costs. The main purpose is we will reduce energy costs from effective maps reduce the concept. In order to optimize the mining results, Evaluate Map Reduce using one step algorithm and three step algorithm Iterative algorithm by various calculations Efficient mining characteristics, too energy.

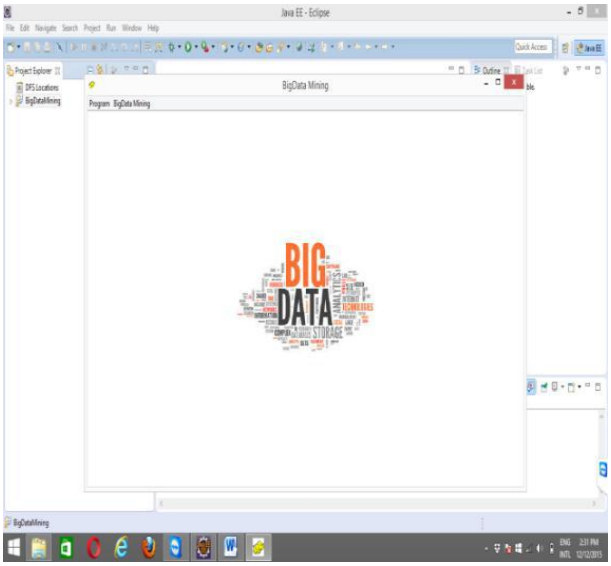


Fig.1 Structure of Big Data

The processing approach called energy map to reduce the scheduling algorithm .EMRSA is an algorithm which provide extra energy and fewer map. Based on priority scheduling is a task to allocate a based on the schedule Trades need and utilization. And map, it is easy because the energy has, to reduce the work of the resource to improve. The final result shows experimental difference it is one of a variety of algorithms included in this paper.

2. RELATED WORKS

2.1 HADOOP:

Hadoop is an open-source framework, which is open source search technology.Hadoop allows to supply and procedure big data in a distributedEnvironment with asymmetrical clusters of computational usingsimple programming models.

It is designed to expand from single server to thousands of machines, each machine contribution local computation and storage [33]. Because of its distributed file system, it can run applications that include thousands of nodes containing terabytes of data [48]. A single node ruin doesn't affect the damaging system failure.

2.2. ENERGY & PERFORMANCE MODELS FOR MAPREDUCE

The user creates the energy and performance models for Map Reduce framework which is used to forecastthe energy

used and presentation of jobs with various Hadoop configuration settings. Theidea to use the multivariate regression modelling on the data collected from the energy reading of the Hadoop Map Reduceto generate these models to control. The parameters added in a model which by getting output by doing the slight factorial analysis of results of the energy description done using the max and min possible values of all the parameters mentioned above. Then make and verify the stochastic Markov chain models for the Map Reduce systems to calculate the performance and energy by making use of data collected from energyrepresentation.

2.3 ENERGY MAP REDUCE SCHEDULING ALGORITHM (EMRSA)

Algorithm 1 EMRSA-X

Create an empty priority queue $\mathcal{Q}^m$
Create an empty priority queue $\mathcal{Q}^r$
for all $j \in \mathcal{A}$ **do**
  $ecr_j^m = \min_{\forall i \in \mathcal{M}} \frac{e_{ij}}{p_{ij}}$, for EMRSA-I; or
  $ecr_j^m = \frac{\sum_{\forall i \in \mathcal{M}} \frac{e_{ij}}{p_{ij}}}{M}$, for EMRSA-II
  $\mathcal{Q}^m$.enqueue(j, ecr_j^m)
for all $j \in \mathcal{B}$ **do**
  $ecr_j^r = \min_{\forall i \in \mathcal{R}} \frac{e_{ij}}{p_{ij}}$, for EMRSA-I; or
  $ecr_j^r = \frac{\sum_{\forall i \in \mathcal{R}} \frac{e_{ij}}{p_{ij}}}{R}$, for EMRSA-II
  $\mathcal{Q}^r$.enqueue(j, ecr_j^r)
$D^m \leftarrow \infty$; $D^r \leftarrow \infty$
while $\mathcal{Q}^m$ is not empty **and** $\mathcal{Q}^r$ is not empty **do**
  $j^m = \mathcal{Q}^m$.extractMin()
  $j^r = \mathcal{Q}^r$.extractMin()
  $f = \frac{\sum_{\forall i \in \mathcal{M}} p_{ij^m}}{\sum_{\forall i \in \mathcal{R}} p_{ij^r}}$
  $\mathcal{T}^m$: sorted unassigned map tasks $i \in \mathcal{M}$ based on p_{ij^m}
  $\mathcal{T}^r$: sorted unassigned reduce tasks $i \in \mathcal{R}$ based on p_{ij^r}
  if $\mathcal{T}^m = \emptyset$ **and** $\mathcal{T}^r = \emptyset$ **then break**
  ASSIGN-LARGE()
  ASSIGN-SMALL()
  if $D^m = \infty$ **then**
    $D^m = D - p^r$
    $D^r = p^r$
if $\mathcal{T}^m \neq \emptyset$ **or** $\mathcal{T}^r \neq \emptyset$ **then**
  No feasible schedule
  return
Output: X, Y

This involves the input files with the .arff extensions, that is, attribute relation file format (ARFF).
Hadoop plug-in is applied in the eclipse environment. Hadoop is the flexible and available architecture for large scale calculation on data processing on a network of service hardware. Eclipse is an integrated development environment (IDE) which contains a base workspace and an extensible plug-in system for customizing the environment. Here in this paper user can able to implementing hadoopplug-in by including the jar files in eclipse and which generates a virtual memory of 1GB.

2.4 ENERGY EFFICIENT CLASSIFICATION METHOD

An efficient organization methodResults evaluated using the following twoOrganization method:

(I) Support Vector Machine (SVM)
(II) Naïve Bayesian.

(I) Support Vector Machine (SVM):

Support vector machine organization method. Supervised Used in the algorithm:

• Classification and regression (binary and multi-classproblem)
• Anomaly detection (one class problem)

The SVM training algorithm, categorize the new example into one category, Non stochastic binary linear classifier.
In the SVM model, Points in space, isolated categories are. It is as wide as possible.Maintenance vector machines are being developed as robust.

Tools for noisy complex classification andregressiondomain. Two important functions of maintenance vector machine It is generality theory, which leads to the principle method to chosen hypothesis and, OS functions, which introduce a non-linearity to the hypothesis space. Explicitly requires a nonlinear algorithm.

2.4.1 SUPPORT VECTORS

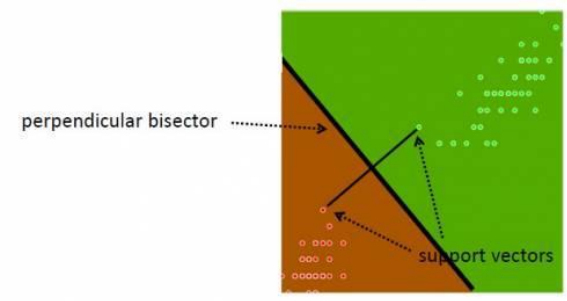


Fig 2. Dimensional Hyper Plain

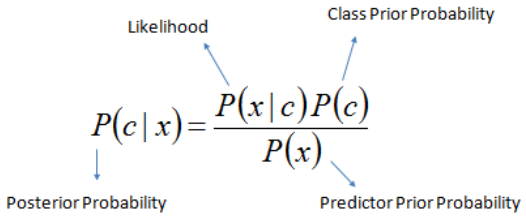
A black line separating the two clouds in the class is in the middle of the channel. Departure is 2D, A Line, 3D is a plane, and over 4 dimensions are hyper planes. Mathematically, separation can be found by taking two critical members, one for each class. These points are called maintenance vectors. These are important points defining the channel. This is the perpendicular bisector of the straight line connecting these two support vectors. It is a concept of maintenance vector machine.

Since SVM is based on a consistent statistical basis and mathematical basis for simplification and optimization theory, it is not classified as a "just another algorithm" class. In addition, it shows that it is superior to the existing technology on various problems in the real world. Although SVM does not solve all of user problems, the kernel method and the maximum margin method are further improved and, when adopted by the data mining community, become an important tool in data minor toolkits.

2.4.2 NAÏVE BAYESIAN

The Naive Bayesian classifier is based on Bayes' theorem with independence supposition between analysts. The Naive Bayesian model is easy to hypothesis, requires no estimation of complex iteration parameters and is particularly useful for very large data sets. Although its simplicity Naive Bayesian classifiers often operate astonishingly well and are widely used as they are often superior to more cultured classification methods. Algorithm:

The Bayes Theorem provides a way to calculate the posterior probability P (c | x) from P (c), P (x), and P (x | c). The Naive Bayes classifier assumes that the effect of the value of analyst (x) on a given class (c) is independent of the values of the other predictors. This supposition is called class conditional independence.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Formula 1. Conditional Probability

- P (c | x) is the subsequent probability of the class (target) given the predictor (attribute).
- P (c) is the preceding probability of the class.
- P (x | c) is the likelihood that the prediction class is a given probability.
- P (x) is the predictor's probability.

Example:

A posterior probability first, each attribute for the target. It can be planned by building aoccurrence table. After that, convert the occurrence table to the likelihood table, and finally calculate the subsequent probability of each class using the naive Bayesian expression. The class with the highest a subsequent probability is the result of the prediction.

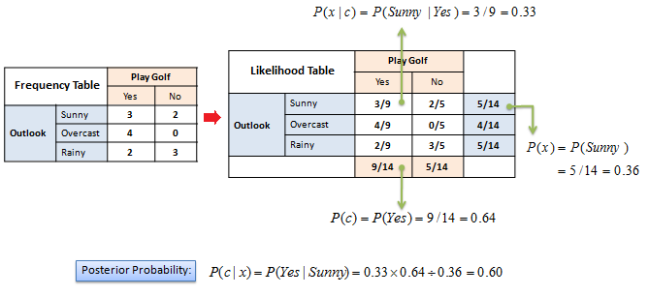


Table 1. Conditional Probability

Zero-frequency problem

If the attribute value (Outlook = Overcast) does not occur with all class values (Play Golf = no), add 1 to the number of all attribute values - class combination (Laplace estimator). Numerical Prognostic Variable Numerical tables must be converted to resounding variables (binning) before creating the frequency table. Another option user have is to use the distribution of numerical variables to get common guesses. For example, one common approach is to assume a normal distribution of numeric variables. The probability density function for a normal distribution is distinct by two limitations (mean and standard deviation).

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{Mean}$$

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5} \quad \text{Standard deviation}$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad \text{Normal distribution}$$

Formula 2. Normal Distribution

2.5.1 MAP REDUCE ARCHITECTURE

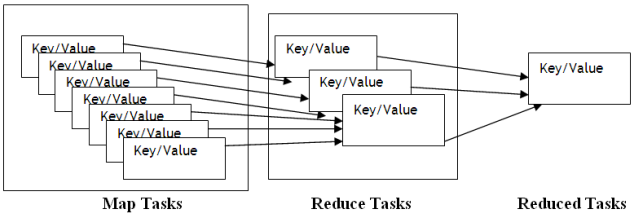


Fig.3 Implementation for Processing and Generating Dataset

Map Reduce is a programming model and relatedapplication for processing and generating large datasets consuming parallel distributed algorithms on clusters. Map Reduce is the center of Hadoop. This programming paradigm enables huge scalability across hundreds or even thousands of servers in a Hadoop cluster. The first map job is a map job that takes a series of data and converts each element into another data set that breaks down into individual dictionaries (key / value pairs). The reduce job accepts the output from the map as input and syndicates the data tuples into a smaller set of As the arrangement named Map Reduce shows, the diminish job always runs after the map job.

2.5.2 SCHEDULING

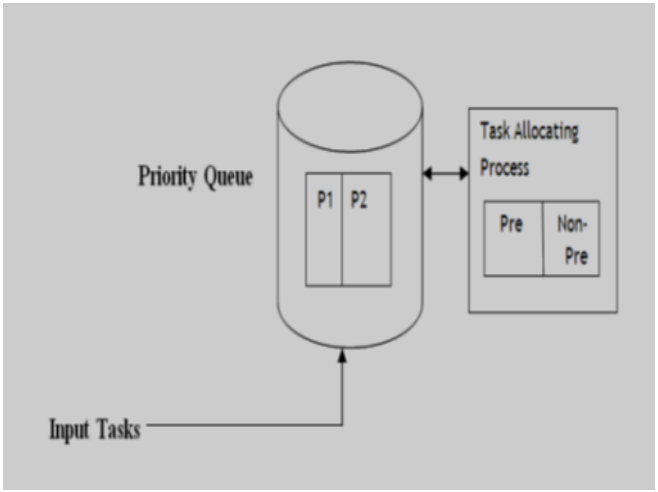


Fig. 4 Scheduling Process

Scheduling process scheduling is afundamental part of the multiprogramming operating system. Such an operating system allows multiple processes to be overloaded into executable memory at one time and the overloaded process parts the CPU using time multiplexing. Priority scheduling. The basic idea is simple. Priority is allocated to each process and priority is executed. Equal priority processes are

scheduled in FCFS order. The shortest job priority (SJF) algorithm is a superior case of the general priority scheduling algorithm.

5. CONCLUSION:

This chapter, and the classification method of maintenance vector machines and naive Bayes for effective data analysis results, said for a set of efficient techniques for repetitive repetition calculation. In a real-time experiment, the described classification method and EMRSA is, suggestively reducing the amount of time it takes in order to refresh the large amounts of data mining results, compared with the re-calculation of a simple replication Map Reduce, consistent efficient Energy use.

REFERENCES

[1] S. Lloyd, "Least squares quantization in PCM," IEEE Trans. Inform. Theory., vol. 28, no. 2, pp. 129–137, Mar. 1982.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in Proc. 20th Int. Conf. Very Large Data Bases, 1994, pp. 487–499.

[3]S. Brin, and L. Page, "The anatomy of a large-scale hypertextual web search engine," Comput. Netw. ISDN Syst., vol. 30, no. 1–7, pp. 107–117, Apr. 1998.

[4] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in Proc. 6th Conf. Symp. Opear. Syst. Des. Implementation, 2004, p. 10.

[5] R. Power and J. Li, "Piccolo: Building fast, distributed programs with partitioned tables," in Proc. 9th USENIX Conf. Oper. Syst. Des. Implementation, 2010, pp. 1–14.

[6] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: A system for large-scale graph processing," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2010, pp. 135–146.

[7] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, "Haloop: Efficient iterative data processing on large clusters," in Proc. VLDB Endowment, 2010, vol. 3, no. 1–2, pp. 285–296.

[8] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: A runtime for iterative mapreduce," in Proc. 19th ACM Symp. High Performance Distributed Comput., 2010, pp. 810–818.

[9] D. Peng and F. Dabek, "Large-scale incremental processing using distributed transactions and notifications," in Proc. 9th USENIX Conf. Oper. Syst. Des. Implementation, 2010, pp. 1–15.

[10] D. Logothetis, C. Olston, B. Reed, K. C. Webb, and K. Yocum, "Stateful bulk processing for incremental analytics," in Proc. 1st ACM Symp. Cloud Comput., 2010, pp. 51–62.

[11] J. Cho and H. Garcia-Molina, "The evolution of the web and implications for an incremental crawler," in Proc. 26th Int. Conf. Very Large Data Bases, 2000, pp. 200–209.

[12] C. Olston and M. Najork, "Web crawling," Found. Trends Inform. Retrieval, vol. 4, no. 3, pp. 175–246, 2010.

[13] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin, "Incoop: Mapreduce for incremental computations," in Proc. 2nd ACM Symp. Cloud Comput., 2011, pp. 7:1–7:14.

[14] Y. Zhang, Q. Gao, L. Gao, and C. Wang, "Priter: A distributed framework for prioritized iterative computations," in Proc. 2nd ACM Symp. Cloud Comput., 2011, pp. 13:1–13:14.

[15] T. J€org, R. Parvizi, H. Yong, and S. Dessloch, "Incremental recomputations in mapreduce," in Proc. 3rd Int. Workshop Cloud Data Manage., 2011, pp. 7–14.

[16] Y. Zhang, Q. Gao, L. Gao, and C. Wang, "imapreduce: A distributed computing framework for iterative computation," J. Grid Comput., vol. 10, no. 1, pp. 47–68, 2012.

[17] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for, in-memory cluster computing," in Proc. 9th USENIX Conf. Netw. Syst. Des. Implementation, 2012, p. 2.

[18] S. R. Mihaylov, Z. G. Ives, and S. Guha, "Rex: Recursive, deltabased data-centric computation," in Proc. VLDB Endowment, 2012, vol. 5, no. 11, pp. 1280–1291.

[19] Y.Zhang,Q.Gao,L.Gao,andC.Wang,"Acceleratelarge-scaleiterative computation through asynchronous accumulative updates," inProc.3rdWorkshopSci.CloudComput.Date,2012,pp.13–22.

[20] C. Yan, X. Yang, Z. Yu, M. Li, and X. Li, "IncMR: Incremental data processing based on mapreduce," in Proc. IEEE 5th Int. Conf. Cloud Comput., 2012, pp.pp. 534–541.

[21] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed graphlab: A framework for machine learning and data mining in the cloud," in Proc. VLDB Endowment, 2012, vol. 5, no. 8, pp. 716–727.

[22] S. Ewen, K. Tzoumas, M. Kaufmann, and V. Markl, "Spinning fast iterative data flows," in Proc. VLDB Endowment, 2012, vol. 5, no. 11, pp. 1268–1279.

[23] D. G. Murray, F. McSherry, R. Isaacs, M. Isard, P. Barham, and M. Abadi, "Naiad: A timely dataflow system," in Proc.24th ACM Symp. Oper. Syst. Principles, 2013, pp. 439–455.

[24] U. Kang, C. Tsourakakis, and C. Faloutsos, "Pegasus: A peta-scale graph mining system implementation and observations," in Proc. IEEE Int. Conf. Data Mining, 2009, pp. 229–238.