Map Reduce Performance in Delay Scheduling for Achieving Locality and Fairness Using Cluster Scheduling

R.Devi Bala[1],G.Pushpa Antanet Sheepa[2]

[1]R.Devi Bala,M.E/CSE,GanadipthyTulsi's Jain Engineering College,Vellore India-632102.

Susendren29@gmail.com

[2]G.Pushpa Antanet Sheepa,Asst Prof/CSE,GanadipathyTulsi's Jain Engineering College,Vellore India-632102.

Pusphaantanetsheepa_cse@grtec.ac.in

ABSTRACT

Map Reduce is programming tool for Hadoop cluster. While allocating resources, Map Reduce has two levels: Task-level and Phase-level. These levels should be used to check performance of each job. There is a limitation with allocating resources at Task-level. So it affects data-locality of a particular job. We present algorithm called PRISM: which presents at the Phase-level. It is called as Phase-level scheduling. In the Phase-level, when we want to schedule a job for the given various resource requirements. So here we find that, PRISM achieves data locality in variety of clusters. This scheduling algorithm may improves execution of one server that is connected to many node it is also called as parallelism, and also improves resource consumption with respect to time. This algorithm is only applicable in the running time of hadoop schedulers. Running time of job is 1.3 time faster than current hadoopscheduler.We first demonstrate the importance of phase-level scheduling by showing the resource usage variability within the lifetime of a task using a wide-range of Map Reduce jobs. We then present a phase-level scheduling algorithm that improves execution parallelism and resource utilization without introducing stragglers. In a 10-node Hadoop cluster running standard benchmarks, PRISM offers high resource utilization and provides 1:3 improvement in job running time compared to the current Hadoop schedulers.

Keywords—PRISM, Hadoop Scheduler utilization, High resource.

1.INTRODUCTION

With the development of the information technology, thes scales of data are increasing quickly. In recentyears, the amount of data in our world has been increasing explosively, and analysing large data sets so called "Big Data" becomes a key basis of competition underpinning new waves of productivity growth, innovation and consumer surplus.Big data is becoming an integral part of solving the world's problems.

1.1 Challenges in Massive Data

The massive data poses a great challenge for classification. Designing the traditional machine learning algorithms for classification of features with MapReduce programming framework is very necessary in dealing with massive datasets. There are so many machine learning algorithms many features.It is widely recognized that a large number of features can adversely affect the performance. Feature Selection is a process of identifying known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context.

Feature selection techniques are a subset of the more general field of feature extraction. Feature extraction creates new existing for good subset of feature selection (classification). Here Fast clustering based feature subset selection is considered for efficient feature selection.

1.2 Map Reduce

Map Reduce, characterized by its remarkable simplicity, fault tolerance, and scalability is becoming a spopular programming framework to automatically parallelize large scale data processing in web indexing, data mining and bio informatics. It is extremely powerful and runs fast for various application areas.

2.FEATURE SELECTION

Subset of most useful features with respect to target concepts.In machine learning and statistics, feature selection, also For high dimensional data, there are so

features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features. Feature selection is useful as part of the data analysis process, shows which features are important for prediction, and how these features are related.With such an aim of choosing a subset of good features with respect to the target concepts, feature subset selection is an effective way for reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. Irrelevant features, along with redundant features, severely affect the accuracy. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Moreover, “good feature subsets contain features highly correlated with the class.”

2.1 Generalization Based Clustering

In cluster analysis, graph-theoretic methods have been well studied and used in many applications. The general graph-theoretic clustering is simple: compute a neighbourhood graph of instances, then delete any edge in the graph that is much longer/shorter (according to some criterion) than its neighbours. The result is a forest and each tree in the forest represents a cluster. In our study, graph-theoretic clustering methods are applied to features. In particular, minimum spanning tree (MST)-based clustering algorithms is adopted, because they do not assume that data points are grouped around centers or separated by a regular geometric curve and have been widely used in practice.

2.2 Organization of the report

Discusses about the literature survey emphasizing the research activities and related works in Texture analysis and Local Binary Pattern. Chapte presents the existing system and its drawbacks. The proposed system and its advantages are also discussed.

3.OVERVIEW

3.1On k-Anonymity and the Curse of Dimensionality

"Big data" warrants innovative processing solutions for a variety of new and existing data to provide real business benefits. But processing large volumes or wide varieties of data remains merely a technological solution unless it is tied to business goals and objectives. In recent years, the wide availability of personal data has made the problem of privacy preserving data mining an important one. A number of methods have recently been proposed for privacy preserving data mining of multidimensional data records. One of the methods for privacy preserving data mining is that of anonymization, in which a record is released only if it is indistinguishable from k other entities in the data. We note that methods such as k-anonymity are highly dependent upon spatial locality in order to effectively implement the technique in a statistically robust way. In high dimensional space the data becomes sparse, and the concept of spatial locality is no longer easy to define from an application point of view. In this paper, we view the k-anonymization problem from the perspective of inference attacks over all possible combinations of attributes. We show that when the data contains a large number of attributes which may be considered quasi-identifiers, it becomes difficult to anonymize the data without an unacceptably high amount of information loss. This is becausean exponential number of combinations of dimensions can be used to make precise inference attacks, even when individual attributes are partially specified within a range.

We provide an analysis of the effect of dimensionality on k-anonymity methods. We conclude that when a data set contains a large number of attributes which are open to inference attacks, we are faced with a choice of either completely suppressing most of the data or losing the desired level of anonymity. Thus, this paper shows that the curse of high dimensionality also applies to the problem of privacy data mining.

3.2 Practical Privacy: The SULQ Framework

The popularity of the Web and Internet commerce provides many extremely large datasets from which information can be gleaned by data mining. This book focuses on practical algorithms that have been used to solve key problems in data mining and which can be used on even the largest datasets. It begins with a discussion of the map-reduce framework, an important tool for parallelizing algorithms automatically. The authors explain the tricks of locality-sensitive hashing and stream processing algorithms for mining data that arrives too fast for exhaustive processing. The PageRank idea and related tricks for organizing the Web are covered next. Other chapters cover the problems of finding frequent itemsets and clustering. The final chapters cover two applications: recommendation systems and Web advertising, each vital in e-commerce.

Written by two authorities in database and Web technologies, this book is essential reading for students and practitioners alike.We consider a statistical database in which a trusted administrator introduces noise to the query responses with the goal of maintaining privacy of individual database entries. In such a database, a query consists of a pair (S, f) where S is a set of rows in the database and f is a function mapping database rows to {0, 1}. The true answer is P i∈S f(di), and a noisy version is released as the response to the query. Results of Dinur, Dwork, and Nissim show that a strong form of privacy can be maintained using a surprisingly small amount of noise – much less than the sampling error – provided the total number of queries is sublinear in the number of database rows. We call this query and (slightly) noisy reply the SuLQ (Sub-Linear Queries) primitive.

The assumption of sublinearity becomes reasonable as databases grow increasingly large. We extend this work in two ways.

First, we modify the privacy analysis to real-valued functions f and arbitrary row types, as a consequence greatly improving the bounds on noise required for privacy. Second, we examine the computational power of the SuLQ primitive. We show that it is very powerful indeed, in that slightly noisy versions of the following computations can be carried out with very few invocations of the primitive: principal component analysis, k means clustering, the Perceptron Algorithm, the ID3 algorithm, and (apparently!) all algorithms that operate in the in the statistical query learning model.

3.3Fuzzy c-Means Algorithms for Very Large Data

Re-identification is a major privacy threat to public datasets containing individual records. Many privacy protection algorithms rely on generalization and suppression of "quasi-identifier" attributes such as ZIP code and birthdate. Their objective is usually syntactic sanitization: for example, k-anonymity requires that each "quasi-identifier" tuple appear in at least k records, while l-diversity requires that the distribution of sensitive attributes for each quasi-identifier have high entropy. The utility of sanitized data is also measured syntactically, by the number of generalization steps applied or the number of records with the same quasi-identifier. In this paper, we ask whether generalization and suppression of quasi-identifiers offer any benefits over trivial sanitization which simply separates quasi-identifiers from sensitive attributes. Previous work showed that k-anonymous databases can be useful for data mining, but k-anonymization does not guarantee any privacy. By contrast, we measure the tradeoff between privacy (how much can the adversary learn from the sanitized records?) and utility, measured as accuracy of data-mining algorithms executed on the same sanitized records.

For our experimental evaluation, we use the same datasets from the UCI machine learning repository as were used in previous research on generalization and suppression. Our results demonstrate that even modest privacy gains require almost complete destruction of the data-mining utility. In most cases, trivial sanitization provides equivalent utility and better privacy than k-anonymity, l-diversity, and similar methods based on generalization and suppression. Very large (VL) data or big data are any data that you cannot load into your computer's working memory. This is not an objective definition, but a definition that is easy to understand and one that is practical, because there is a dataset too big for any computer you might use; hence, this is VL data for you. Clustering is one of the primary tasks used in the pattern recognition and data mining communities to search VL databases (including VL images) in various applications, and so, clustering algorithms that scale well to VL data are important and useful. This paper compares the efficacy of three different implementations of techniques aimed to extend fuzzy c-means (FCM) clustering to VL data. Specifically, we compare methods that are based on 1) sampling followed by noniterative extension; 2) incremental techniques that make one sequential pass through subsets of the data; and 3) kernelized versions of FCM that provide approximations based on sampling, including three proposed algorithm.

3.4 Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge

In the Social Web, a number of diverse recommendation approaches have been proposed to exploit the user generated contents available in the Web, such as rating, tagging, and social networking information. In general, these approaches naturally require the availability of a wide amount of these user preferences. This may represent an important limitation for real applications, and may be somewhat unnoticed in studies focusing on overall precision, in which a failure to produce recommendations gets blurred when averaging the obtained results or, even worse, is just not accounted for, as users with no recommendations are typically excluded from the performance calculations. In this article, we propose a coverage metric that uncovers and compensates for the incompleteness of performance evaluations based only on precision. We use this metric together with precision metrics in an empirical comparison of several social, collaborative filtering, and hybrid recommenders.

The obtained results show that a better balance between precision and coverage can be achieved by combining social-based filtering (high accuracy, low coverage) and collaborative filtering (low accuracy, high coverage) recommendation techniques. We thus explore several hybrid recommendation approaches to balance this trade-off. In particular, we compare, on the one hand, techniques integrating collaborative and social information into a single model, and on the other, linear combinations of recommenders. For the last approach, we also propose a novel strategy to dynamically adjust the weight of each recommender on a user-basis, utilizing graph measures as indicators of the target user's connectedness and relevance in a social network. Privacy is an important issue in data publishing.

Many organizations distribute non-aggregate personal data for research, and they must take steps to ensure that an adversary cannot predict sensitive information pertaining to individuals with high confidence. This problem is further complicated by the fact that, in addition to the published data, the adversary may also have access to other resources (e.g., public records and social networks relating individuals), which we call external knowledge. A robust privacy criterion should take this external knowledge into consideration.

In this paper, we first describe a general framework for reasoning about privacy in the presence of external knowledge. Within this framework, we propose a novel multidimensional approach to quantifying an adversary's external knowledge. This approach allows the publishing organization to investigate privacy threats and enforce privacy requirements in the presence of various types and amounts of external knowledge. Our main technical contributions include a multidimensional privacy criterion that is more intuitive and flexible than previous approaches to modeling background knowledge.

In addition, we provide algorithms for measuring disclosure and sanitizing data that improve computational efficiency several orders of magnitude over the best known techniques.

3.5 Revealing Information while Preserving Privacy

We examine the tradeoff between privacy and usability of statistical databases. We model a statistical database by an n-bit string d1,..,dn, with a query being a subset $q \subseteq [n]$ to be answered by Σiεq di. Our main result is a polynomial reconstruction algorithm of data from noisy (perturbed) subset sums. Applying this reconstruction algorithm to statistical databases we show that in order to achieve privacy one has to add perturbation of magnitude (Ω√n). That is, smaller perturbation always results in a strong violation of privacy. We show that this result is tight by exemplifying access algorithms for statistical databases that preserve privacy while adding perturbation of magnitude Õ(√n).For time-T bounded adversaries we demonstrate a privacy preserving access algorithm whose perturbation magnitude is ≈ √T.

4. EXISTING SYSTEM

Distributed reasoning methods on computing RDF closure for reasoning, which takes much time (usually several hours or even days for large) and space (generally the ontology size is more than original data size).Moreover, each time when new RDF arrives, full reasoning over the entire dataset is needed to compute the new RDF closure. This process occurs at every which is too time-consuming in practice. Web PIE newly-arrived RDF triples and old ones but fails consider the relations between them, thus resulting in a number of duplicated triples during the reasoning thereby its performance. A centralized architecture executed on a single machine or local server when dealing with large datasets, distributed reasoning approaches executed on multiple computing nodes have thus emerged to improve the scalability and speed of inferences.

Disadvantages of the existing system the data volume of RDF closure is ordinarily larger than original RDF data.The storage of RDF closure is thus not a small amount and the query on it takes nontrivial time.The data volume increases and the ontology base is updated, these methods require the recomputation of the entire RDF closure every time when new data arrive.Which takes much time (usually several hours or even days for large) and space (generally the ontology size is more thanoriginal data size).

5. PROPOSED SYSTEM

Recommender systems (RSs) are techniques and intelligent applications to assist users in a decision making process where they want to choose some items among a potentially overwhelming set of alternative products or services. Collaborative filtering (CF) such as item- and user-based methods are the dominant techniques applied in RSs. The basic assumption of user-based CF is that people who agree in the past tend to agree again in the future. Different with user-based CF, the item-based CF algorithm recommends a user the items that are similar to what he/she has preferred before. Consequently, service recommendation based on the similar users or similar services would either lose its timeliness or couldn't be done at all.

Advantages of the proposed systemofLinear scalability, automatic failover support, and convenient backup of Map Reduce jobs.Distributed data on the web make it difficult to acquire appropriate triples for appropriate inferencesWhich can well leverage the old and new data to minimize the updating time and reduce the reasoning time when facing big RDF datasets.To speed up the updating process with newly-arrived data and fulfill the requirements of end-usersFor online queries we show that slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of ℓ-diversity.We develop an efficient algorithm for computing the sliced table that satisfies ℓ-diversity. Our algorithm partitions attributes into columns, applies column generalization, and partitions tuples into buckets. Attributes that are highly-correlated are in the same column.We conduct extensive workload experiments.

6. DESIGN

In many cases, such delays can also propagate to phases in other tasks, causing them to be delayed as well. For example, even though the execution of a shuffle phase of a reduce task can overlap with the execution of a merge phase of a map task, the shuffle phase cannot finish unless all merge phases of the map tasks have finished.

Thus, when choosing between scheduling merge phases and shuffle phases, it is preferable to give sufficient resources to merge phases to allow them to finish faster, instead of allocating most of the resources to the shuffle phase and delay the completion of merge phases. The above examples demonstrate the importance of achieving a fairness-performance trade-off for phase scheduling. In other words, it is necessary to provide fairness without significantly delaying the execution of each phase. Given a set of phases that can be scheduled on a machine, the scheduler assigns a utility value to each phase which indicates the benefit of scheduling the phase. The scheduler will then schedule the phases in decreasing value of their utility. If a phase is map or shuffle, scheduling the phase implies scheduling a new map or reduce task. In this case, the utility of the phase is determined by the increase in parallelism from running an additional task. For other phases, the utility is deter- mined by the urgency to complete the phase. A simple met- ric for measuring urgency is the number of seconds that a task has been paused due to phase-level scheduling. If the task has been paused for a long time, it becomes urgent to schedule its remaining phases in order to avoid creating a staggler.

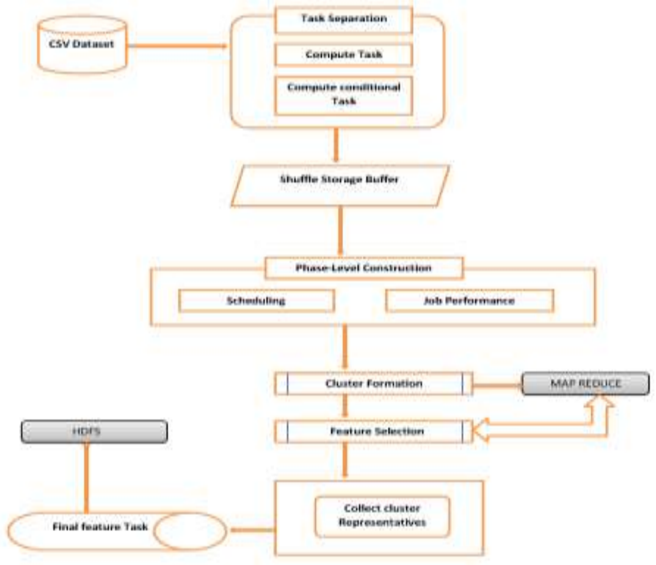

fig:Framework of our architecture.

7. EXPERIMENT

7.1 Original Data

We conduct extensive workload experiments. Our results confirm that slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. In some classification experiments, slicing shows better performance than using the original data.

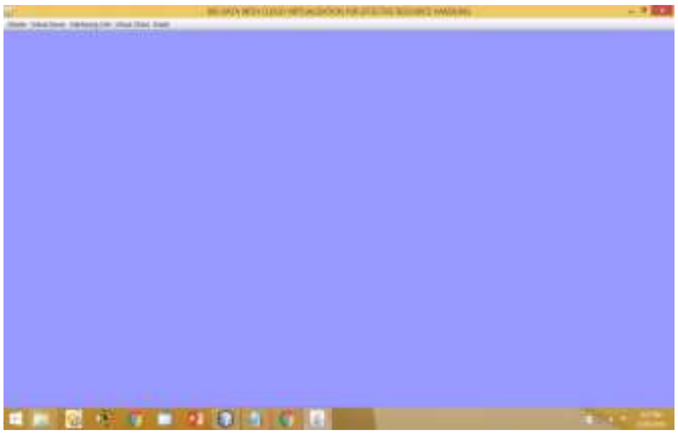

Fig1.Original Data.

7.2 Generalized Data

Fig2.Generalized Data.

Generalized Data, in order to perform data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized interval/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data.

7.3 Bucketized Data

we show the effectiveness of slicing in membership disclosure protection. For this purpose, we count the number of fake tuples in the sliced data. We also compare the number of matching buckets for original tuples and that for fake tuples. Our experiment results show that bucketization does not prevent membership disclosure as almost every tuple is uniquely identifiable in the bucketized data.

A FAST CLUSTERING FEATURES BASED ON SUB SELECTION ALGORITHM IN BIG ... VIRTUAL SERVER
Virtual Client A
Virtual Client B
Virtual Client C
Virtual Client D

Fig3.Bucketized Data.

7.4 Multiset-based Generalization Data

We observe that this multiset-based generalization is equivalent to a trivial slicing scheme where each column contains exactly one attribute, because both approaches preserve the exact values in each attribute but break the association between them within one bucket.

Fig4.Multiset-based Generalization Data.

7.5 One-attribute-per-Column Slicing Data

We observe that while one-attribute-per-column slicing preserves attribute distributional information, it does not preserve attribute correlation, because each attribute is in its own column. In slicing, one groups correlated attributes together in one column and preserves their correlation.

Fig5.One-attribute-per-Column Slicing Data

For example in the sliced table shown in Table correlations between Age and Sex and correlations between Zipcode and Disease are preserved. In fact, the sliced table encodes the same amount of information as the original data with regard to correlations between attributes in the same column.

7.6 Sliced Data

Another important advantage of slicing is its ability to handle high-dimensional data. By partitioning attributes into columns, slicing reduces the dimensionality of the data.

Fig6.Sliced Data.

Each column of the table can be viewed as a sub-table with a lower dimensionality. Slicing is also different from the approach of publishing multiple independent sub-tables in that these sub-tables are linked by the buckets in slicing.

8.CONCLUSION

This work motivates several directions for future research in Big Data using HDFS.First, in this paper, we consider slicing where each attribute is in exactly one column. An extension is the notion of overlapping slicing, which duplicates an attribute in more than one columns. This releases more attribute correlations. For example, in Table, one could choose to include the Disease attribute also in the first column. That is, the two columns are {Age,Sex,Disease} and {Zipcode,Disease}. This could provide better data utility, but the privacy implications need to be carefully studied and understood. It is interesting to study the tradeoff between privacy and utility.Second, we plan to study membership disclosure protection in more details. Our experiments show that random grouping is not very effective. We plan to design more effect tive tuple grouping algorithms.Third, slicing is a promising technique for handling high-dimensional data.By partitioning attributes into columns, we protect privacy by breaking the association of uncorrelated attributes and preserve data utility by preserving the association between highly-correlated attributes. For example, slicing can be used for anonymizing transaction databases, which has been studied recently.

Finally, while a number of anonymization techniques have been designed, it remains an open problem on how to use the anonymized data. In our experiments, we randomly generate the associations between column values of a bucket. This may lose data utility. Another direction to design data mining tasks using the anonymized data computed by various anonymization techniques.

9.FUTURE WORK

This project is implemented using Internet Information Services, ASP.net and HADOOP. This chapter provides an insight into each of these that are used in our system and concludes with a few snapshots of the final GUI. The corrective maintenance is to correct the diagnosis errors in this project. It helps to easily identify the diagnosis errors in this software for debugging it. It modifies this project with a changing environment. It recommends for new capabilities in any module, modifications to the existing functions and increases the value of this project i.e., general enhancements. Preventive Maintenance or Reengineering takes much time. It costs significant amount of money. It absorbs resources that might be otherwise occupied on immediate concerns. For all these reasons, reengineering is not accomplished in a few months or even a few years. That's why every organization needs a pragmatic strategy for software reengineering.

REFERENCES

1. A Fast clustering Based Feature Subset Selection Algorithm for High Dimensional Data,IEEE,2013, QinBao Song,Jingjie Ni and Guangato Wang.

2. A. Arauzo-Azofra, J.M. Benitez, and J.L. Castro, "A Feature Set Measure Based on Relief," Proc. Fifth Int'l Conf. Recent Advances in Soft Computing, pp. 104-109, 2004.

3. A Scalable Random Forest Algorithm Based on MapReduce ,2013 IEEE Jiawei Hanl, YanhengLiul, Xin Sunl.

4 .A Survey on Feature Selection Using FAST approach to reduce High Dimensional Data, International Journal of Engineering Trends and Technology, Feb 2014,S.Saranya.

5. R. Butterworth, G. Piatetsky-Shapiro, and D.A. Simovici, "On Feature Selection through Clustering," Proc. IEEE Fifth Int'l Conf.Data Mining, pp. 581-584, 2005.

6. A. Rasmussen, M. Conley, R. Kapoor, V. T. Lam, G. Porter, and A. Vahdat, "ThemisMR: An I/O-Efficient MapReduce," in Proc. ACM Symp. Cloud Comput., 2012, p. 13.

7. [17] A. Verma, L. Cherkasova, and R. Campbell,"Resource provi- sioning framework for MapReduce jobs with performance goals," in Proc. ACM/IFIP/USENIX Int. Conf. Middleware, 2011, pp. 165–186.

8. D. Xie, N. Ding, Y. Hu, and R. Kompella, "The only constant is change: Incorporating time-varying network reservations in data centers," in Proc. ACM SIGCOMM, 2012, pp. 199–210.

9. Y. Yu, M. Isard, D. Fetterly, M. Budiu, U. Erlingsson, P. Gunda, and J. Currey, "DryadLINQ: A system for general-purpose dis- tributed data-parallel computing using a high-level language," in Proc. USENIX Symp. Oper. Syst. Des. Implementation, 2008, pp. 1–14.

10. .M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in Proc. Eur. Conf. Comput. Syst., 2010, pp. 265–278.