AN EFFECTIVE ADAPTIVE APPROACH FOR JOINING DATA IN DATA WAREHOUSE

Sudha.S[1], Manikandan.S[2]

[1]Research Scholar, Adhiparasakthi Engineering College, Chennai, Tamilnadu, India
[2]Professor, Computer Applications, RMD Engineering College, Chennai, Tamilnadu, India

Abstract

Formulation of efficient assessment is important for the businesses, because its retrieve lot of details from the data warehouse. In Data warehouses have materialize as original business information pattern where data store and maintain in concurrent. The adaptations are requiring in the implementation of Extract Transform Load (ETL) operations. The several methods are included to joining stream and produce the innovative relation .The previous work was used the adaptive join in data warehouse using ETL procedure. This approach was common conspire, which create many possible solution. But the drawback of the previous approach is its not consider exact reproduction. To rise above the question, we are going to present genetic algorithm for joining stream of data . Several queries and streams are combined in data warehouse the selection of exact grouping of multiple associations are complete via genetic algorithm. The crossover as well as mutation prefer the paramount consortium of several associations of link for by retrieving the data and produces the output in data warehouse. The performance of the proposed genetic algorithm used to deliver efficient highest join data and increase the scalability.

Keywords: *join, stream, relation, Genetic algorithm.*

---***---

1. INTRODUCTION

Different way we have to store the data in different database. Genetic Algorithms are powerful search techniques used to solve many difficult problems. Despite the great successes achieved in real-world applications, GAs have some drawbacks. In the genetic algorithm lot of fitness calculations are necessaryed before an acceptable solution can be found. Fitness evaluation is not easy in many real-world applications. There are several situations, in which the fitness evaluation becomes computationally difficult, so, GAs can be very demanding in terms of computation and GAs are frequently used to solve search and optimization problems since they were first introduced by Holland [1]. They have gained this prominence by their robustness and simplicity they offer. Individuals with higher aptitude have more 26 probability to survive, to reproduce, and to transmit their genetic characteristics to future generations. GAs can perform efficient search operations in problem spaces where it is not easy to understand the environment. Each potential solution in the search space is considered as an individual (phenotype). Individuals are represented by using strings that are called chromosomes. Genes are the atomic parts of chromosomes and they codify a specific characteristic of a chromosome. There are several approaches to encode individuals for a variety of applications. GAs generate a an initial population at the first phase of the algorithm and then, selection (for mating and carrying its genes to the next generation, giving higher probability to "fitter" individuals representing better solutions) crossover (method for combining genes of two mating parents), and mutation operations are applied randomly on the current generation, creating the next generation of solutions [3]. Individual having the best fitness in the population is the proposed solution of the problem. For each pair of mating individuals, the parents' chromosomes are split in two (or more) parts and genes selected from both parents are combined to generate a new chromosome which joins the population. Mutations are also possible where an individual's randomly selected gene is mutated. Mutations prevent stagnation and enable a wider exploration of search space that could not be reached with crossover operators as it is limited with the available genes in the current population [2]. In order to keep the population size constant a method is defined for selecting the individuals that will be copied to the next generation, and another iteration begins. Termination can be based on either by production of a fixed number of generations or the algorithm can terminate when the amount of improvement in the overall generation quality (e.g. average fitness values for individuals in a generation) falls below a predetermined threshold. Also, GA can be terminated at any given time reporting the best currently available individual as the discovered solution, thus making GA very suitable for real-time problems where only a fixed amount of time is available for optimization.

After an introduction ETL and Genetic Algorithm in Section1, related literature review of the joining algorithms are given in Section 2. Section 3 defines the main process models and the mutation and offspring structure . Evaluation of performance and the related results are presented in Section 4. Concluding clarification and the future works are given in Section 5.

2. RELATED WORKS

Hash-Merge Join (HMJ) (Mokbel et al. 2004), also one from the series of symmetric joins, is based on push technology and consists of two phases, hashing and merging. All three approaches above do not consider the metadata about the stream. Therefore, they are unable to identify the data which is no-longer required and by plummeting it the overhead can be reduced. In addition, the join approaches above focus on throughput optimization while ignoring the other optimization goals such as the characteristics of stream data which are correspondingly important.

The MESHJOIN (Mesh Join) algorithm (Polyzotis et al. 2007) (Polyzotis et al. 2008) has been presented with the objective to amortize the slow disk access with as many stream tuples as possible. To perform the join, the algorithm keeps a number of hunks of stream in memory at the same time. In each iteration the algorithm loads a disk partition into memory and attains the join with all these stream chunks. The algorithm performs tuning for efficient memory transfer among the join components, but It is identified in the past some issues around the access to the disk based relation. Also MESHJOIN cannot deal with intermittency of the stream competently.

R-MESHJOIN (reduced Mesh Join) (Naeem et al. 2010) is an enhanced form of MESHJOIN in which one issue related to suboptimal distribution of memory among the join components is resolved. However, R-MESHJOIN implements the same strategy as the MESHJOIN algorithm for retrieving the disk-based relation.

A partition-based approach (Chakraborty et al. 2009) has been introduced to deal with intermittency in the stream. It uses a two-level hash table to attempt to join stream tuples as soon as they arrive, and uses a partition-based waiting area for the other stream tuples. The authors do not provide a cost model for their approach. In addition, the algorithm needs a clustered index or an equivalent sorting on the join attribute and it does not prevent starvation of stream tuples.

One recent algorithm, HYBRIDJOIN (Hybrid Join) (Naeem et al. 2011) address the issue of retrieving the disk-based relation. An active strategy to access the disk-based relation is introduced in HYBRIDJOIN.

Another advantage of HYBRIDJOIN is that it can deal with burst streams, which is a curb of both MESHJOIN and R-MESHJOIN. However, if it is consider long-tail distributions, it is find that the algorithm can be improved further.

3. SYSTEM MODEL

The proposed work used Genetic Algorithm is designed here to perform the multi-join operation efficiently in active data warehouse. Normally, in active data warehouse, the information should keep up-to-date with recent values in the database. So, to add the recent values with active data warehouse, the genetic algorithm is used here by performing the highest joining operation in the source streams. The architecture diagram of the proposed work to perform the highest joining operation using GA is shown in Fig 1

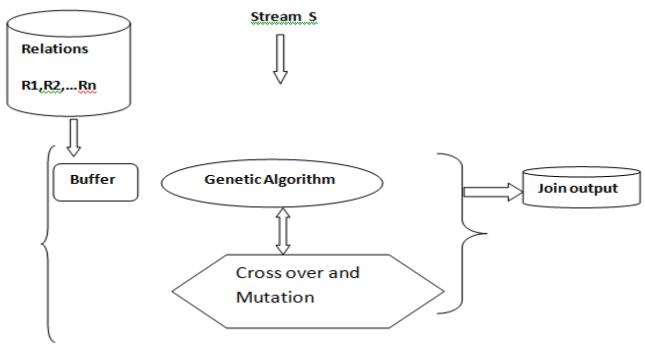


Fig. 1 System Architecture Diagram of highest joining operation using Genetic Algorithm

In this system model we can insert the current data to active data warehouse, the GA use to verify the exact data and add it in to the warehouse. so this model explain task to execute the highly join using GA. The main component of the figure .1. are crossover estimation and mutation, while the source stream S and relation Rare the input. We first select the relation R based on the source stream and perform the highly join operation using Genetic Algorithm process.

Initial relations are loaded in the relation table, the quality of relations are evaluated based on the fitness .so the initial relation store in the relation table.

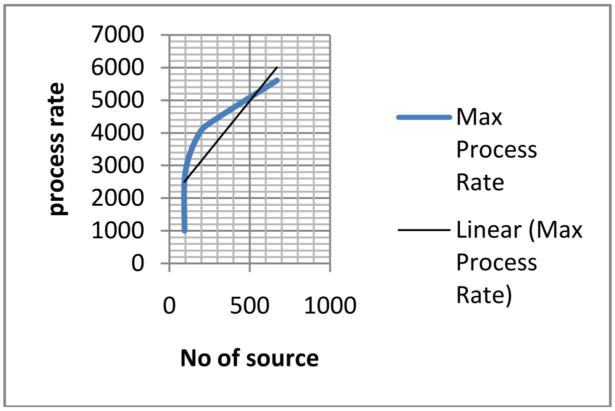
4. PERFORMANCE EVALUATION

The performance of GA based highly join operation in data warehouse are considered the following ,Data retrieval , Efficiency of join operation , Scalability

Table 1

No of Source	Max Process Rate
95	1000
100	2800
189	4000
300	4450
670	5600

Table 1 given details about the number of source arrival in the method and maximum process rate. Based on this its fashioned the following presentation.



5. CONCLUSIONS

In this paper, we propose a genetic algorithm based to improve the performance of our existing join data. This work carry out highly join operation with relation table in limited remembrance by using genetic algorithm. An efficient way to retrieve the data using this GA,GA effectively perform highly join data by selection ,crossover and mutation.

REFERENCES

[1]. Holland, J.H. Adaptation in Natural and Artificial Systems.University of Michigan Press, 1975, Ann Arbor, MI, USA.

[2]. Sevinc, E., and Cosar, A. An Evolutionary Genetic Algorithm for Optimization of Distributed Database Queries. The Computer Journal, vol.54, issue: 5, 2011, 717-725.

[3]. Swami, A., and Gupta, A. Optimization of large join queries.Proceedings of ACM SIGMOD Conf. on Management of Data, Chicago, Ill, May,1988, 8–17.

[4]. S.Sudha and S.Manikandan," ADAPTIVE APPROACH FOR JOINING AND SUBMISSIVE VIEW OF DATA IN DATA WAREHOUSE USING ETL", Indian Journal of Computer Science and Engineering (IJCSE), Vol. 4 No.3 Jun-Jul 2013 Pp.250-252

[5]. Naeem, M. A., Dobbie, G. & Weber, G. (2011), 'HYBRIDJOIN for Near-real-time Data Warehousing', International Journal of Data Warehousing and Mining (IJDWM), IGI Global.

[6]. Naeem, M. A., Dobbie, G. & Weber, G. (2011), X-HYBRIDJOIN for Near-real-time Data Warehousing, in 'Proceedings of 28th British National Conference on Databases (BNCOD '11)', Springer, Manchester, UK, pp. 33–47.

BIOGRAPHIES

Sudha.S received the Master Degree in Computer Applications from the Bharathidhasan University, Trichy, India in the year 2000, and the M.Phil degree from the same University in the year 2003.

She is teaching profession for the past 13 years. Previously she worked as a Lecturer in the Department of M.S and B.Teh(IT). She is currently an Assistant Professor; Adhiparasakthi Engineering College affiliated to Anna University, Chennai, India from 2003 to till. She is currently perusing here PhD degree in Department Computer Science, Anna University, Chennai, India. She has published 2 papers in refereed journals and 5 papers national and international conference proceedings.

Dr. S. Manikandan received B.Sc. degree in Mathematics from Madurai Kamaraj University, Madurai in 1996 and M.C.A. degree from Bharathidasan University, Tiruchirapalli, in 1999. He received his M.Phil. degree in Computer Science from Manonmaniam Sundaranar University, Tirunelveli, in 2003 and Ph.D degree from Anna University Chennai in the year 2009.

He is teaching profession for the past 11 years. Previously he worked as an Assistant Professor, PSNA Engineering college, Dindigul. He is currently Director in the Department of Computer Applications RMD engineering College, Chennai .He has published six research articles in International and National journals and presented twenty papers in refereed national & International Conferences.