Journal of Computer Sciences and Applications, 2015, Vol. 3, No. 6, 130-133
Available online at http://pubs.sciepub.com/jcsa/3/6/4

DOI:10.12691/jcsa-3-6-4

Performance Evaluation of Complex Data Sets with Heterogeneity Using Particle Swarm Optimization

Mishra Jyoti Prakash, Mishra Sambit Kumar*

Gandhi Institute for Education and Technology, Baniatangi, Bhubaneswar, Odisha, India
*Corresponding author: sambitmishra@gietbbsr.com

Abstract Traditional query processing applications may not be adequate with large or complex data sets with heterogeneity. Challenges to this context may include analysis, capture, search, sharing, storage, transfer, visualization, and information privacy. Cloud computing refers to the practice of transitioning computer services such as computation or data storage to multiple redundant offsite locations available on the internet, that allows application software to be operated using internet enabled devices. Cloud computing usually focuses on maximizing the effectiveness of the shared resources. Cloud resources are generally not only shared by multiple users but are dynamically reallocated as per demand. The present cloud services realize improved execution efficiency by aggregating application execution environments. Now a day it is in the phase of expanding from application aggregation and sharing data aggregation and utilization. In this paper, the query evaluation strategies have been proposed by considering partially correlated data in heterogeneous databases of concern. The main idea behind this strategy is to retrieve the data from heterogeneous databases linked with the declarative query I interface implementing data access methods and optimization mechanisms. The indexing and query processing strategies may be applied to the integrated components of the database systems with heterogeneity. As a result, it may be convenient and useful to analyze and evaluate the data using efficient functional evaluations implemented inside the database systems. Usually the index structures are generated to coordinate the result analysis without duplicating the query evaluation result. It is also aimed to provide an end-to-end solution for scalable access to big data integration, where end users may formulate queries based on a familiar conceptualization of the domain. It has also been proposed to process the distributed query in the heterogeneous environment to evaluate scalable solution for executing queries in the cloud.

Keywords: *query plan, cloud, big data, tuple, aggregation, temporal data, swarm, pbest, gbest*

Cite This Article: Mishra Jyoti Prakash, and Mishra Sambit Kumar, "Performance Evaluation of Complex Data Sets with Heterogeneity Using Particle Swarm Optimization." *Journal of Computer Sciences and Applications*, vol. 3, no. 6 (2015): 130-133. doi: 10.12691/jcsa-3-6-4.

1. Introduction

The query rewriting process may be evaluated effectively and efficiently against the integrated data sources including temporal data and data streams. This efficiency for big data is very much required for executing database queries. The plans associated with the queries in the heterogeneous database environment may be achieved to measure the efficiency by either running queries with the maximum amount of parallelism at each stage of execution or by allowing to execute the same query with the use of resources that depends on the resource availability for the particular query.

Usually the queries may be expressed in structured query language. The initial phase of query optimization is planning. The result of this phase is the database query strategies. A database partition may be categorized as a set of tuples having a particular property e.g. the value of a hash function applied on one column of the database may be the same for all the tuples in the same partition. So a database/ relation may be defined as a set of partitions. As a result, the optimizer may produce an execution plan may be needed to execute the query. The resources needed to execute the queries may be allocated automatically. These resources may be transferred to the locations used to extract the details from the physical machine or a virtual machine in a cloud. The queries may be efficiently processed and expressed in a declarative language. It is understood that when a query is received, one or more optimization engines produce an execution plan that may contain the resulted sequence of operators and the data partition upon which they must be applied. The optimization engines report back to the main server which then utilizes the execution engines in coordination with the clients to execute the query. In this case some clients need to communicate with external databases.

The present cloud services have remarkable improvements on execution strategies by aggregating application execution environments which may be in the phase of expanding from application aggregation and sharing to data aggregation and utilization.

Big data processing on cloud may include more entities that may lead to generalization of sizable amount of read and write requests. Therefore it is required to put more

number of servers for data storage to uniformly distribute the loads. The processing state of the query depends on the order of arrival of the query as well as operation results. To ensure the order of arrival of query, the scheduler schedules the execution mechanisms and sends instructions to the individual clients to synchronize the entire processing tasks.

2. Review of Literature

Google Prediction API et.al [1] allows users to create machine learning models to predict numeric values for a new item based on values of previously submitted training data or predict a category that best describes an item. The prediction API allows users to submit training data as comma separated files following certain conventions, create models, share their models or use models that others shared.

L. Proctor et. al [2] in their paper have defined that Cloud-enabled Big Data analytics poses several challenges with respect to replicability of analyses. When not delivered by a Cloud, analytics solutions are customer-specific and models often have to be updated to consider new data.

Centola D et. Al [3] in their paper have discussed that the data-intensive applications commonly require significant storage space and intensive computing power. The demand of such resources alone, however, is not the only fundamental challenge of dealing with big data.

Wu X et.al [4] have described in their paper that the data storage and management systems should support high throughput data access with millions of tweets generated each second, though the tweets may be generated from different geographic regions.

Hess B et.al [5] have focused in their paper about simulation of data that is typically stored in data files, which are further organized into various levels of directories. Data access is enabled by encoding the descriptions of the content in files into the names of files and directories, or storing more detailed descriptions about the file content in separate metadata files.

Ivanova M et.al [6] in their work have focused on extending the kernel functionalities of DBMSs to meet challenges in scientific data management. It includes work that deals with query language, data storage, data compression, index design, I/O scheduling, and data provenance.

Feig M et.al [7] have focused in their work about analytical query that deals with mathematical function and maps the readings of a group of atoms to a scalar, vector, a matrix, or a data cube. For the purpose of studying the statistical feature of the system, popular queries in this category may include density, first-order statistics, second-order statistics, and histograms.

Guo, Yubin et.al [8] have discussed in their work that data owners may outsource their query services and data, but data is very sensitive and private assets of them and it should be protected from the service provider and the querying users in some extent. Data owner might be update, query and authorize access on the data, while the service providers in cloud should know nothing about especially detailed data about data.

TingjianGe et.al [9] have focused in their paper about user privacy and data privacy. They have discussed how to enforce data privacy and user privacy over outsourced database service.

Hu, Haibo et.al [10] have described in their paper that query users need to query and exact data from cloud, but the query might disclose some sensitive information, behavior patterns of the user. Privacy of data owners and query users are defined as data privacy and user privacy respectively.

Jeong H, Park J et.al [11] have discussed about cloud in their paper that provides computing and storage services to users via the Internet. Public clouds offer these services to both organizations and individuals, but require no infrastructure or maintenance investment. Therefore, more applications and services are expected to rely on cloud resources in the future.

A, Katal et.al [12] in their work have focused on Big Data that is used to describe massive volumes of structured and unstructured data .It is very difficult to process this data using traditional databases and software technologies believed to be originated from the Web search companies who had to query loosely structured very large distributed data.

F.C.P et.al [13] in their work focused on big data application that refers to the large scale distributed applications deals with large data sets. Data exploration and analysis turned into a difficult problem in many sectors in the span of big data. With large and complex data, computation becomes difficult to be handled by the traditional data processing applications which triggers the development of big data applications.

Xu-bin et.al [14] in their work have focused on Bioinformatics which requires a large scale data analysis. Cloud computing gets the parallel distributed computing framework together with computer clusters and web interfaces.

3. Problem Formulation

3.1. Algorithm

Step 1: Assign the size of the swarm, 10
Step 2: Specify dimension of the problem, 30
Step 3: Maximum number of iterations, 100
Step 4: Specify cognitive parameter and determine social parameter
Step 5: Generate random population of continuous values by considering size of the swarm and constriction factor into account
Step 6 : Generate random velocities by considering size of the swarm and along with the dimension
Step 7: For each particle , calculate fitness value
Step 8: If the fitness value is better than the best fitness value (pbest) set current value as the new pbest
Step 9: Choose the particle with the best fitness value of all the particles as the gbest
Step 10: For each particle, calculate particle velocity considering the size of the swarm and constriction factor
Step 11:Update particle position according to the dimension of the problem
Step 12: Initial function evaluation by considering the velocity
Step 13: Evaluate population cost using initial function

Step 14 : Determine the mean cost of query plan
Step 15: Determine the local and global minima (query plan) for each particle
Step 16: Evaluate the cost of local minima (query plan) and find the best particle
Step 17: Update the velocity and evaluate weight index of each particle
Step 18: Update the particle position
Step 19: Update the best local and global position of particle

3.2. Experimental Analysis

No. of processors associated: 20
Maximum number of databases per individual location: 50

The particle swarm optimization (PSO) uses a population of individuals to search feasible region of the solution space. In this context, the population is called swarm and individuals are called particles. It uses number of particles that constitute a swarm. Each particle keeps a track of its coordinates and obtains the best solution pbest. It also keeps track of neighborhood particle and its best value called gbest. PSO is usually a computational method that optimizes the problem by iteratively taking particle's position and velocity and using mathematical formulae. Each particle shows one possible solution of optimization problem.

Table 3.2.1. Selection parameter and weight of the task associated with query plan

Query plan	Tasks associated with query plan	Selection parameter associated with query plan	Weight of the tasks associated with query plan
14	64	52.7178	19.0078
11	62	52.6902	19.0074
28	78	52.7992	19.0093
23	73	52.7873	19.0087
16	66	52.7275	19.0079
24	74	52.7879	19.0088
31	82	52.8114	19.0096

Maximum 50 queries per individual databases were considered. As shown in Table 3.2.1., it is seen that the selection parameter associated with query plan depends on the tasks associated with the query plan.

Table 3.2.2. Location cost and cost of particles associated with query

Location cost of particles (pbest1)	Location cost of particles (pbest2)	Cost of particles (pbest1)	Cost of particles (pbest2)
0.8147	0.1576	1.8480	0.5506
0.9058	0.9706	1.8235	1.9518
0.1270	0.9572	0.5370	2.0066
0.9134	0.4854	2.1381	0.9861
0.6324	0.8003	1.4910	1.6329
0.0975	0.1419	0.4477	0.5583
0.2785	0.4218	0.8047	1.0751

The best location of particles may be retrieved from the location cost of particles along with cost of particles as shown in the Table 3.2.2. In this case the gbest1 values achieved are 0.0975, 0.8147 and the values of gbest2 achieved are 0.1419, 0.1576.

Table 3.2.3. Fitness value and cost of tasks associated with query

Fitness value of tasks (pbest1)	Fitness value of tasks (pbest2)	Selection parameter of tasks associated with query plans	Cost of tasks associated with query plans
0.6557	0.7060	40.7625	0.0250
0.7431	0.6948	69.275	0.7340
0.8491	0.2769	79.93	0.8500
0.9340	0.0462	81.56	0.8750

As shown in Table 3.2.3.it is also seen that the selection parameter of tasks associated with query plans depends on the fitness parameter of the tasks.

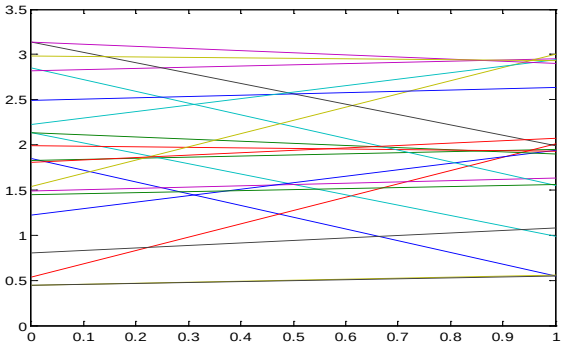

Figure 3.2.1. (Location cost of particles(pbest) VS Cost of (best local position of particles)

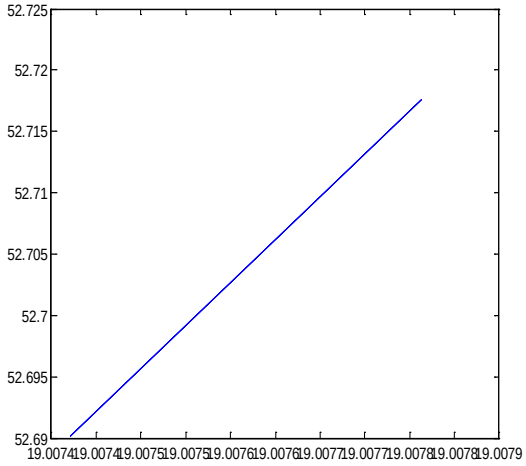

Figure 3.2.2. (Weight of the tasks associated with query plans VS selection parameter associated with query plans)

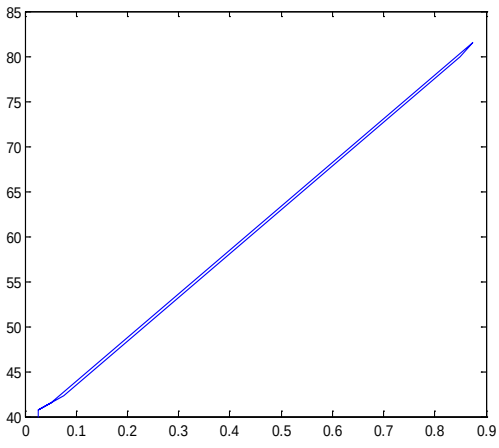

Figure 3.2.3. (Cost of tasks associated with query plans VS selection parameter of tasks of query plans)

4. Discussion and Future Direction

Usually in traditional query processing, the system may have computational as well as communication limitations. But while processing a query in cloud, not only the primary data center but also the cloud may identify the traversed paths and may obtain any information that may point out the query point as the exact distances to the query point. The comparisons and feasibility study have already been done by analyzing the performance of the complex data sets with heterogeneity. It has been observed the improvements on the computation and query response time under various query types and parameter settings. Particle swarm optimization technique may be implemented to solve optimization problems in the presence of a wide range of uncertainties, like processing of noisy data or changing the parameters of design variables after optimization, and the quality of the obtained optimal solution. It has also been observed that the fitness function may suffer from approximation errors. The optimum in the problem space may change over time. The algorithm should be able to track the optimum continuously. The target of optimization may change over time. The demand of optimization may adjust to the dynamical environment. Some additional measures may be considered while implementing swarm optimization techniques that may be able to solve satisfactorily dynamic problems.

5. Conclusion

It is understood that cloud computing helps in storing of data to maximize resource utilization, it is very much essential for the data to be protected with proper authorization. In this case, the machine plays the role of the client to store the sensitive data in the cloud.

Cloud may be intended to optimize systems by identifying valuable information via analysis of aggregated data. Data analysis must be repeated many times from different perspectives and low cost processing may be required in all phases of development and operation. The benefits offered by cloud such as virtually allocation of computational resources may have potential to approach it.

The big data may be created in many fields. With the big data analytics techniques and particle swarm optimization techniques, more applications or systems may be designed to solve real world problems.

References

[1] Google Prediction API, http://developers.google.com.

[2] L. Proctor, C.A. Kielszewski, A Hochstein, Spangler, Proceedings of the Annual SRII Global conference 2011.

[3] Centola D. The spread of behavior in an online social network experiment. Science 329:1194-1197, 2010.

[4] Wu X, Zhu X, Wu G-Q, Ding W Data mining with big data. IEEE Trans Knowl Data Eng 26(1):97-107, 2014.

[5] Hess B, Kutzner C, van der Spoel D, Lindahl E (2008) GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. J Chem Theory Comput 4(3):435-447.

[6] Ivanova M, Kersten ML, Nes N (2008) Adaptive segmentation for scientific databases. In: ICDE. IEEE, Cancún, México. pp 1412-1414.

[7] Feig M, Abdullah M, Johnsson L, Pettitt BM (1999) Large scale distributed data repository: design of a molecular dynamics trajectory database. Future Generation Comput Syst 16(1):101-110.

[8] Guo, Yubin, et al. "A solution for privacypreserving data manipulation and query on nosql database." Journal of Computers 8.6 (2013): 1427-1432.

[9] TingjianGe, Stanley B. Zdonik, and Stanley B. Zdonik. Answering aggregation queries in a secure system model. In VLDB, pages 519-530, 2007.

[10] Hu, Haibo, et al. "Processing private queries over untrusted data cloud through privacy homomorphism." Data Engineering (ICDE), 2011 IEEE 27th International Conference 2011.

[11] Jeong H, Park J An efficient cloud storage model for cloud computing environment. In: Proceedings of international conference on advances in grid and pervasive computing, 2012 vol 7296, pp 370-376.

[12] A, Katal, Wazid M, and Goudar R.H. "Big data: Issues, challenges, tools and Good practices." Noida: 2013, pp. 404-09, 8-10 Aug. 2013.

[13] F.C.P, Muhtaroglu, Demir S, Obali M, and Girgin C. "Business model canvas perspective on big data applications." *Big Data, 2013 IEEE International Conference*, Silicon Valley, CA, Oct 6-9, 2013, pp. 32-37.

[14] Xu-bin, LI, JIANG Wen-rui, JIANG Yi, ZOU Quan "Hadoop Applications in Bioinformatics." Open Cirrus Summit (OCS), 2012 Seventh, Beijing, Jun 19-20, 2012, pp. 48-52.