International Journal of Database Theory and Application
Vol.8, No.5 (2015), pp.149-160
http://dx.doi.org/10.14257/ijdta.2015.8.5.12

Efficient Query Processing Platform for Uncertain Big Data

Zhenhua Huang, Jiawen Zhang and Qiang Fang

Department of Computer Science Tongji University, Shanghai, China
huangzhenhua@tongji.edu.cn

Abstract

Query processing technology has recently received a lot of attention in the business intelligence and information service communities. However, the existing approaches can not efficiently optimize the query performance in the uncertain big data environment. In this paper, we propose QPPUBG, a novel and efficient query processing platform for uncertain big data. QPPUBG mainly includes four modules: (i) query equivalence reconstructing for uncertain big data; (ii) multiple query optimization over probability relation components; (iii) query execution plan constructing over probability relation components, and (iv) physical implementation solution of query for uncertain big data. Specially, QPPUBG can support the possible world instance semantics and efficiently handle arbitrary decision spaces. Moreover, QPPUBG can seamlessly integrate the above four modules into the modern parallel computation frameworks. We present the extensive experiments that demonstrate QPPUBG is both efficient and effective.

Keywords: *big data, query processing, query optimization, parallel computation*

1. Introduction

As the in-depth application of information and network technologies, the data in enterprises increase exponentially. And fast obtaining the useful information by query processing is an important tool to provide efficient decision-making for enterprise users. Recently, the researchers have focus on using online analytical processing (OLAP) and data mining (DM) these two aspects to improve the quality of decision-making [1]. Online analytical processing aims to adopt a series of complex multi-dimensional queries including top-n query, KNN query, rank query, range query, skyline query and iceberg query, to explore and analyze the whole enterprise data and return the profile information from massive data. Then users can complete decision analysis according to small-volume profile information.

During recent two years, big data has recently received a lot of attention in many countries such as America, China, Japan, Britain, and Germany. Specially, American government announce to invest two hundred million dollars for the research plan of big data [3]. The European Commission is funding a 2-year-long big data public private forum through their seventh framework program to engage companies, academics and other stakeholders in discussing big data issues [4]. It is not difficult to find that the age of big data was approaching. From literature [5], we can see that the 4V's of big data – volume, velocity, variety and veracity—makes the query processing challenging for the traditional data analysis technologies. And these 4V characteristics indicate that the data volume is too huge, and data values change too fast, and it does not follow the rules of conventional query processing. Hence, we need to design and develop the new technologies for efficient query processing, and under-stand the problem deeply enough to perform the analytics. Literature [6] proposes sophisticated parallel statistical algorithms for big data, with a focus on density-based methods. Specially, in this paper, the authors implement these algorithms using both map/reduce and SQL interfaces over a variety of storage mechanisms. Literature [7] presents an end-to-end solution for scalable

ISSN: 2005-4270 IJDTA

access to big data integration, were end users will formulate queries based on a familiar conceptualization of the underlying domain. Literature [8] first introduces the Hadoop map/reduce and uses it for query processing big data, and then proposes several different data management techniques, such as job optimization, physical data organization. Literature [9] proposes a HACE theorem that characterizes the features of the big data revolution, and proposes a big data processing model, from the data mining perspective.

As data acquisition requirement of enterprises increases, uncertain big data has received a lot of attention. In most of trades (*e.g.* finance, telecommunication, aero-space), enterprises are restricted by the accuracy of data acquisition devices, and the self-vague of big data, and hence the uncertainty of big data is ubiquitous in data sources of enterprises [10]. Because uncertain big data needs to add the probability distribution information and possible world instance semantics for the objects, query processing over uncertain big data is more complex than certain big data. Thereby, the existing approaches can not efficiently optimize the query performance in the uncertain big data environment.

Motivated by the above facts, in this paper, we first propose QPPUBG (Query Processing Platform for Uncertain Big Data), an efficient platform for query processing and optimization over uncertain big data. Detailed, our QPPUBG platform mainly includes the following four modules: (*i*) query equivalence reconstructing for uncertain big data; (*ii*) multiple query optimization over probability relation components; (*iii*) query execution plan constructing over probability relation components, and (*iv*) physical implementation solution of query for uncertain big data. Specially, our QPPUBG platform can support the possible world instance semantics and efficiently handle arbitrary decision spaces. Furthermore, our QPPUBG platform can seamlessly integrate the above four modules into the modern parallel computation frameworks. We present the extensive experiments that demonstrate that our QPPUBG platform is both efficient and effective.

2. Platform Framework Overview

As the appearance of uncertain big data, existing query processing approaches have serious drawbacks in the aspects of real-timing, robustness and self-adaptability. To solve these main drawbacks, we design and develop QPPUBG, an efficient platform for query processing and optimization over uncertain big data. The platform framework of QPPUBG is shown in Figure 1.

Our QPPUBG platform mainly includes four modules:

2.1. Module 1: Query Equivalence Reconstructing for Uncertain Big Data

The modern storage models of uncertain big data are based on possible world instance semantics [11], there-fore compared with a certain object, they need to store the exponential number of probability relation instances. And for convenient management and query, these storage models store uncertain big data by two stages. In the first stage, various probability instances of the same relation object are organized as a restrictive G-Tabset table $\mathfrak{R}$. In the second stage, $\mathfrak{R}$ is factorized multinomial number of probability relation components, and each probability relation component is the conjunction of several probability relation instances. In many real applications, enterprises usually do not store massive probability relation instances, and only store two equivalent data: (1) the small-scale set W of probability relation components, $W=\{WSD_{1,\ldots,}WSD_n\}$; and (2) the set D of Datalog language rules, $D=\{DL_{1,\ldots,}\ DL_m\}$. Specially, the form of DLi can be denoted as WSD_i:-$Insi_1\wedge\ldots\wedge Insi_x$, which indicates that the probability relation component W is the conjunction of x probability relation instances $Insi_{1,\ldots,}\ Insi_x$.

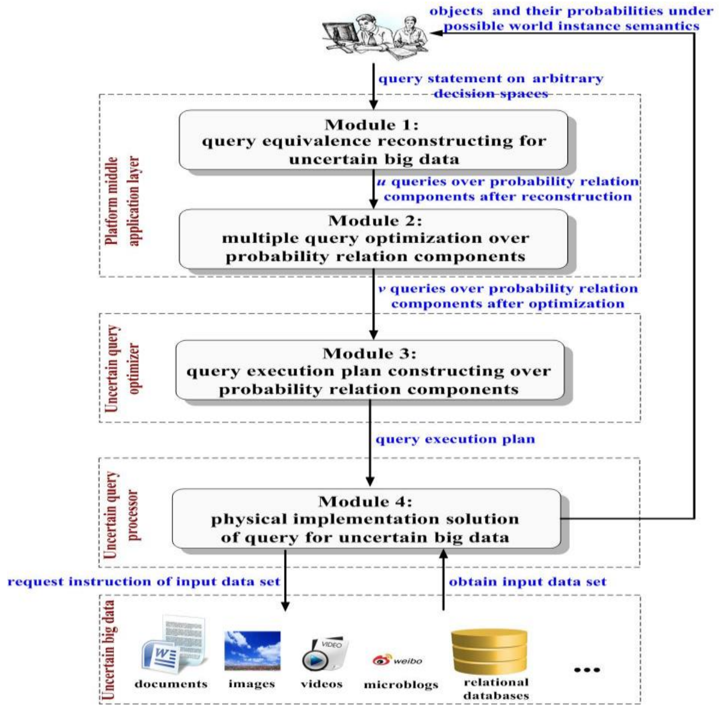
objects and their probabilities under possible world instance semantics
query statement on arbitrary decision spaces
Platform middle application layer
Module 1: query equivalence reconstructing for uncertain big data
u queries over probability relation components after reconstruction
Module 2: multiple query optimization over probability relation components
v queries over probability relation components after optimization
Uncertain query optimizer
Module 3: query execution plan constructing over probability relation components
query execution plan
Uncertain query processor
Module 4: physical implementation solution of query for uncertain big data
request instruction of input data set
obtain input data set
Uncertain big data
documents
images
VIDEO
videos
weibo
microblogs
relational databases
...

Figure 1. The Platform Framework of QPPUBG

In our QPPUBG platform, in order to understand, the input parameter of the proposed query SQ is probability relation instances, *i.e.* SQ:- $\nabla(Ins_1 \wedge \ldots \wedge Ins_x)$. This means that our QPPUBG platform obtains the query result from the input data Ins_1□…□Ins_x. In order to let the query optimizer identify and analyze the query sentence over un-certain big data, in this module, we equally transform SQ to serveral queriers over probability relation components, *i.e.* SQ:- $\nabla(WSD_1) \wedge \nabla(WSD_2) \wedge \ldots \wedge \nabla(WSD_u)$, u≤n.

2.2. Module 2: Multiple Query Optimization over Probability Relation Components

Based on Module 1, our QPPUBG platform has u queries □$\nabla(WSD_1),\ldots, \nabla(WSD_u)$. For each query $\nabla(WSD_i)$, $i \in [0, 1]$, its input parameter is one probability relation component $WSD_i \in W$. In order to return the complete query result, a straightforward method is to respectively obtain the query results of these u probability relation components. However, we find that this straightforward method has two serious performance drawbacks: (1) since the query is CPU-sensitive, this method will spend much CPU time; and (2) since each probability relation component usually occupies massive storage space, it needs much I/O time to put these u probability relation components from disk to the memory.

In order to overcome the above two drawbacks, our QPPUBG platform is based on the cost evaluation and selects the optimal v ($v<u$) WSD_1',…,WSD_v' from the set W of probability relation components $\{WSD_{1,\ldots,}WSD_n\}$. Specially, the query object set of WSD_i' ($1 \le i \le v$) can be used to handle the queries over probability relation components among $WSD_1,\ldots,WSD_u$. Thus, our QPPUBG platform only needs less number of queries and less size of probability relation components.

Furthermore, in our QPPUBG platform, we theoretically prove that compared with the straightforward method, our optimization method can save $1/e \approx 37\%$ CPU time and $(e\text{-}1)/e \approx 63\%$ I/O cost.

2.3. Module 3: Query Execution Plan Constructing Over Probability Relation Components

After our QPPUBG platform achieves v (v<u) probability relation components WSD_1',…, WSD_v', for each WSD_i', before obtaining its query object set, the query optimizer needs to produce efficient query execution plan logically.

For this purpose, in our QPPUBG platform, we design the rule system $\wp$ to equally transform execution order between the query operator and uncertain relation operators (*e.g*. Uselection, Conf, Merge and Ujoin). And we present the cost evaluation of before-transformation and after-transformation. Meanwhile, we theoretically prove the correctness of the rule system $\wp$.

Based on the rule system $\Re$, we modify the left-depth conjunction tree, and thus can improve the query execution plan.

2.4. Module 4: Physical Implementation Solution of Query for Uncertain Big Data

After the query optimizer produces the query execution plan, it submit this query execution plan to the query processor. Then, our QPPUBG platform calls the query processor to physically implement this query execution plan, and obtain the query object set and its existence probability under possible world instance semantics.

We find that it will arise the following two problems if our QPPUBG platform integrates the existing query implementation methods into the query processor:

(1) The existing query implementation methods are only for fixed decisive spaces, and the indexes used in these existing methods, such as R-tree, kd-tree and AR-tree [12], are scalar. The scalar index structures map the multidimensional coordinate space into one-dimensional real number. And hence it loses most location information. Consequently, these existing methods can not be extended to arbitrary decisive spaces.

(2) The existing query implementation methods do not adequately consider the efficiency of the existence probability under possible world instance semantics. And the time complexity of obtaining existence probability for these existing methods is #P-Hard. Hence it is impractical in the real applications.

In our QPPUBG platform, in order to solve the above problems, we use the regular grid index to replace the scalar index structures, and equally transform the computation of existence probability under possible world instance semantics to the computation of the number of true assignments for disjunctive normal form (DNF). The detailed process is seen in Section III.

3. Specific Realization of Our QPPUBG Platform

In this section, we give the specific realization of our QPPUBG platform.

3.1. Realization for Module 1

In the modern enterprises, the storage form of uncertain big data is the probability relation components *W* and Datalog language rules *D*. However, for the proposed query *SQ*, its input parameter is probability relation in-stances. Hence, in order to produce efficient query execution plan, our QPPUBG platform needs to equally reconstruct *SQ* to several queries over probability relation components. For realize this task, we solve the following two difficult points: (*i*) the time complexity of equal reconstruction, and (*ii*) the designing of efficient equal re-construction algorithm.

For the first difficult point, our QPPUBG platform uses the Datalog tool and the first order predicate logic as the reconstruction description language, and defines the formalized semantics for the query of uncertain big data. Then, based on the reconstruction semantics of the first order predicate logic, our QPPUBG platform describes the extensional condition which the query needs to be satisfied. And according to this extensional condition, our QPPUBG platform proposes the decidability proof of query equal reconstruction, the reconstruction time complexity of including/excluding negative predicate, the reconstruction time complexity of including/excluding HAVING clause, the reconstruction time complexity of including/excluding arithmetic predicate, the reconstruction time complexity of including/excluding CONF clause, and the reconstruction time complexity under multi-set semantics.

For the second difficult point, our QPPUBG platform realizes the equal reconstruction algorithm through two phases. In the first phase, our QPPUBG platform is based on the Datalog language rules D, and uses the inversion rules technology [13] to filter the probability relation components M which are irrelevant to the query in polynomial time complexity. And in the second phase, our QPPUBG platform first confirms the minimal number u of probability relation components used for equal reconstruction. Then for the probability relation components W-M, our QPPUBG platform takes the predicate isomorphism as the filtering characteristics, and utilizes the Apriori property [14] to obtain all the candidate probability relation components $H_1,\ldots, H_\zeta$ whose cardinal numbers equal u. Finally, our QPPUBG platform selects arbitrary one set H_i ($i\in[0, 1]$) of candidate probability relation components, and uses H_i as the output result of the equal reconstruction algorithm.

3.2. Realization for Module 2

After implementing the query equal reconstruction, our QPPUBG platform has u queries $\nabla(WSD_1),\ldots, \nabla(WSD_u)$. For each query $\nabla(WSD_i)$, $i\in[0, 1]$, its input parameter is one probability relation component $WSD_i\in W$. In Module 2, the task of our QPPUBG platform is to obtain the query result sets from these u probability relation components.

In order to complete the above task, our QPPUBG platform first constructs the query cost model, which needs to adequately consider the distribution characteristics of uncertain big data. We propose two different technologies to construct the query cost model:

(*i*) In the first technology, our QPPUBG platform uses the query optimizer of existing uncertain database systems to collect statistical informations of the query periodically. The statistical informations include multidimensional index structures of the original data set, the combined probability function, the probability density function, and the frequency of repeated values of uncertain objects, *etc*. And then our QPPUBG platform utilizes the APA1+ sampling estimator [15] whose precision bound is $e/(e + 0.5)\approx 84.3\%$ to locally sample the statistical informations, and approximates the query cost model of the whole original data set according to the one on the small sample data set.

(*ii*) In the second technology, our QPPUBG platform is based on the Monte Carlo method [16], and uses the VG dynamic generator to produce the interval histogram for each dimension in the decision space, then obtains the probability density function of uncertain data set. Next, our QPPUBG platform presents the multiple integral expressions with a rigorous correctness proof. Specially, in the multiple integral expressions, the main body of integrand is the probability density function, and the integrating range is the value range of uncertain data on the decision space.

Based on the query cost model, our QPPUBG platform then proposes an efficient method to select the optimal v ($v<u$) WSD_1',…,WSD_v' from the set W of probability

relation components {$WSD_1,\ldots,WSD_n$}. Specially, the query object set of WSD_i' ($1\leq i\leq v$) can be used to handle the queries over probability relation components among $WSD_1,\ldots,WSD_u$. The detailed process can be described as follows:

(*i*) When u is smaller (≤ 6), our QPPUBG platform usually spends less time cost to complete the task of query optimization. In this case, we first construct the weighted directed bipartite graph, and map probability relation components W and S into the set of nodes. Meanwhile, based on the query cost model, we map the cost among probability relation components into the sets of nodes and edges. And then, we equally transform the multi-query optimization problem on probability relation components to the minimum weighted set cover problem, and obtain the exact v optimal probability relation components.

(*ii*) According to the graph theory [17], the minimum weighted set cover problem is NP-complete time complexity. Hence when u is larger (>6), the scale of the weighted directed bipartite graph will rapidly increase. This makes our QPPUBG platform spend more time cost to complete the task of multi-query optimization. In this case, we utilize the shortest path optimization theory of graph [18], and introduce a virtual vertex, then transform the weighted directed bipartite graph to Steiner weighted path graph [19] in constant time complexity. Finally, we present an efficient method which has the optimization lower-bound guarantee, and produce the Steiner tree from Steiner weighted path graph in polynomial time complexity. Based on the Steiner tree, we obtain the approximate optimal solution of multi-query optimization. According to the directed Steiner tree theory, the time complexity and the optimization lower-bound of our approximate method can be adjusted and balanced by a positive which is no less than 1.

3.3. Realization for Module 3

Obtaining efficient query execution plan has two difficult points: (*i*) design the correct rule system $\wp$ to equally transform execution order between the query operator and uncertain relation operators (*e.g.* Uselection, Conf, Merge and Ujoin); (*ii*) obtaining efficient query execution plan according to the rule system $\wp$.

For the first difficult point, our QPPUBG platform proposes the operation laws (such as the commutative law, combined law, grouping law and duplicate elimination law, *etc*.,) which are satisfied between the query operator and different uncertain relation operators. And our QPPUBG platform supports the equal transformation between different operation execution orders. Meanwhile, we are based on the multi-set theory and the first-order predicate logic, and proves the correctness an completeness of the rule system $\wp$.

For the second difficult point, our QPPUBG platform takes the simple query execution plan provided by the query optimizer as the basic point. And on the left-depth conjunction tree, our QPPUBG platform uses the different equivalence transformation rules to produce different candidate operation execution sequences through the strategy of pushing up/down operation nodes, merging/splitting operation nodes, and transforming operation nodes, *etc.* Then based on the query cost model, our QPPUBG platform computes the time cost for each candidate sequence, and select the optimal sequence to produce the query execution plan with the least time cost. In order to efficiently reduce the number of candidate sequences, our QPPUBG platform balances the time cost of searching optimal execution plan and the benefit of implementing this execution plan, and proposes a heuristic algorithm to fast produce the quasi-optimal query execution plan with the optimization lower-bound guarantee.

3.4. Realization for Module 4

In order to physically implement the query execution plan on uncertain big data, we need to complete the following two tasks: (*i*) efficiently organize and index uncertain relation objects, and can fast obtain the query objects set on arbitrary decision space; and (*ii*) efficiently compute the existence probability under possible world instance semantics for query objects set.

For the first task, our QPPUBG platform designs the regular grid index structure to organize and index uncertain relation objects on arbitrary decision space (shown in Figure 2). And then our QPPUBG platform obviously de-creases the time cost of obtain the query objects set through two phases: (*i*) in the first phase, our QPPUBG platform is based on the minimal description length rule [20] and automatically deletes possible world instances in the regular grid which the users are not interested in; (*ii*) in the second phase, our QPPUBG platform uses the domination and mutex relationships between cells in the regular grid to reduce the comparison number between possible world instances. Furthermore, in order to save the storage space of the index structure, our QPPUBG platform presents the interval dynamic partitioning technique for each dimension in the regular grid, and balances the quantitative distribution of possible world instances in each cell.

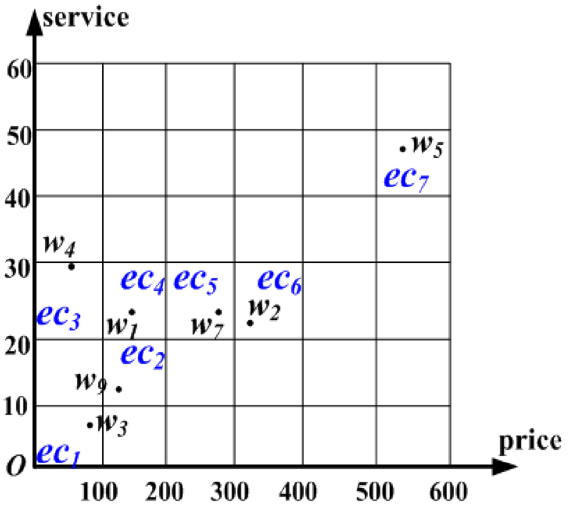

Figure 2. The Regular Grid Index Structure

For the second task, our QPPUBG platform equally transforms the computation of existence probability under possible world instance semantics to the computation of the number of true assignments for disjunctive normal form (DNF), and designs two different methods to solve this task. The first method uses the Davis-Putnam function [21] to exactly obtain the number of true assignments for DNF. Given a DNF normal form, our QPPUBG platform uses the Davis-Putnam function to transform sub-DNF normal forms which are independent and do not share variables, and then exactly obtains the number of true assignments by the recursive way. The second method uses the Karp-Luby random algorithm [22] to the approximation of true assignments with the precision lower-bound guarantee in polynomial time. Specially, the Karp-Luby random algorithm is based on the idea of Monte Carlo, and confirms the approximation of true assignments through N-step stochastic simulation.

4. The Advantages of our QPPUBG Platform

Our QPPUBG platform has the following five advantages:

(1) As for the uncertain big data, our QPPUBG platform uses the probability relation components W and the Datalog language rules D to equally reconstruct the query of probability relation instances. Further, Our QPPUBG platform proposes the efficient equal reconstruction algorithm, and presents the reconstruction time complexity under multi-set semantics.

(2) Our QPPUBG platform adequately considers the distribution characteristics of uncertain big data, and adopts two different technologies to construct the query cost model. Then based on the query cost model, our QPPUBG platform obtains the query execution plan in the query optimizer.

(3) Our QPPUBG platform equally transforms the problem of the multi-query optimization on probability relation components to the minimum weighted set cover problem of bipartite graph and the problem of directed Steiner tree, and separately obtain the exact optimal and quasi-optimal solutions.

(4) For the uncertain big data, our QPPUBG platform designs the correct rule system $\wp$ to equally transform execution order between the query operator and uncertain relation operators (*e.g.* Uselection, Conf, Merge and Ujoin). And then by using the rule system $\wp$, our QPPUBG platform extends the left-depth conjunction tree, and obtains the efficient query execution plan. Specially, the correctness and completeness of the rule sys-tem $\wp$ is through strict theoretical proof.

(5) On the physical level, for different decision demands, Our QPPUBG platform designs the query implementation technology for arbitrary decision spaces. Meanwhile, based on the Davis-Putnam function and the Karp-Luby random algorithm, Our QPPUBG platform obtains the exact and approximate values of query objects set under possible world instance semantics.

5. Experimental Evaluation

This section conducts an empirical study of our QPPUBG platform using the benchmark synthetic da-tasets DATA_1, DATA_2, and DATA_3. The DATA_1 dataset has 107 tuples, and the number of dimensions varies in the range [2, 10]. The DATA_2 dataset has 8 dimensions, and the number of tuples varies in the range [2×10^7, 10^8]. In the DATA_3 dataset, the number of tuples varies in the range [10^8, 5×10^8], and the number of dimensions varies in the range [5, 25]. We evaluate the efficiency and the scalability of our QPPUBG platform.

In the first group of experiments, we use the DATA_1 dataset. Figure 3 shows the experimental results for this group.

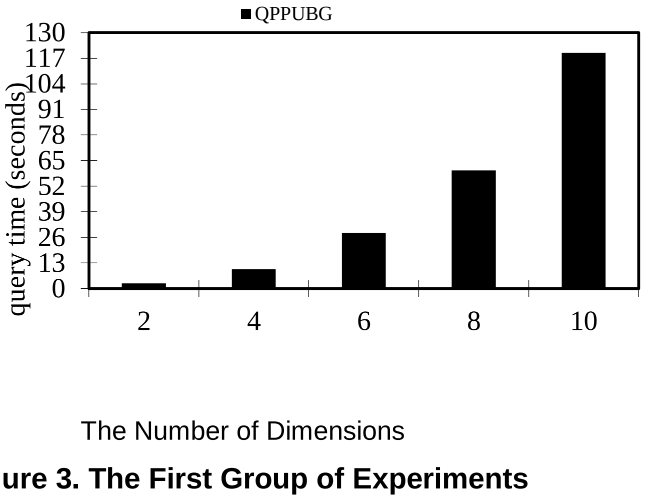

Figure 3. The First Group of Experiments

In the Figure 3, we can observe that our QPPUBG platform has the good query performance and high extendibility. For example, in Figure 3, when the number of dimensions of the DATA_1 dataset equals 2, the query time of our QPPUBG platform is equal to 2.69 seconds. And when the number of dimensions of the DATA_1 dataset equals 10, the query time of our QPPUBG platform is equal to 119.64 seconds.

In the second group of experiments, we use the DATA_2 dataset. Figure 4 shows the experimental results for this group.

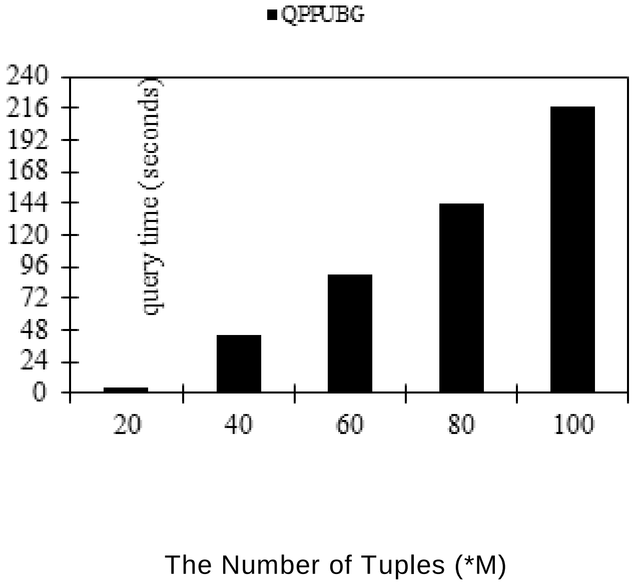

Figure 4. The Second Group of Experiments

In the Figure 4, like the first group of experiments, we can observe that our QPPUBG platform has the good query performance and high extendibility. For example, in Figure 4, when the number of tuples of the DATA_2 dataset equals 2×10^7, the query time of our QPPUBG platform is equal to 4.53 seconds. And when the number of tuples of the DATA_2 dataset equals 10^8, the query time of our QPPUBG platform is equal to 216.96 seconds.

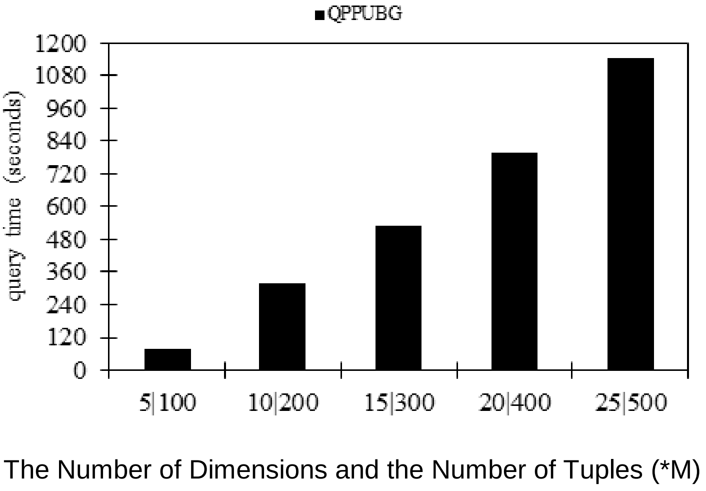

Figure 5. The Third Group of Experiments

In the third group of experiments, we use the DATA_3 dataset. Figure 5 shows the experimental results for this group.

In the Figure 5, like the first two groups of experiments, we can observe that our QPPUBG platform has the good query performance and high extendibility. For example, in Figure 5, when the number dimensions and the number of tuples of the DATA_3 dataset equal 5 and 10^8 respectively, the query time of our QPPUBG platform is equal to 79.34 seconds. And when the number dimensions and the number of tuples of the DATA_3 dataset equal 25 and 5×10^8 respectively, the query time of our QPPUBG platform is equal to 1142.17 seconds.

6. Conclusions

Query processing on uncertain big data is an important problem in the business intelligence and information service communities. In order to improve the query performance and extendibility, we design and develop the QPPUBG platform which mainly includes four closely related modules: (*i*) query equivalence reconstructing for uncertain big data; (*ii*) multiple query optimization over probability relation components; (*iii*) query execution plan constructing over probability relation components, and (*iv*) physical implementation solution of query for uncertain big data. Specially, QPPUBG can support the possible world instance semantics and efficiently handle arbitrary decision spaces. Moreover, QPPUBG can seamlessly integrate the above four modules into the modern parallel computation frameworks. We present the extensive experiments that demonstrate QPPUBG is both efficient and effective.

Acknowledgments

This work was supported in part by a grant from the National Natural Science Foundation of China (No. 61272268), the Program for New Century Excellent Talents in University (NCET-12-0413), the Fok Ying-Tong Education Foundation (142002), National 973 Program (2014CB340404).

References

[1] C. Doulkeridis and K. Nørvåg, "A survey of large-scale analytical query processing in MapReduce", The VLDB Journal, vol. 23, no. 3, **(2014)**, pp. 355-380.

[2] S. Viglas, G. Bierman and F. Nagel, "Processing declarative queries through generating imperative code in managed runtimes", IEEE Data Eng. Bull., vol. 37, no. 1, **(2014)**, pp. 12-21.

[3] R. Compton, C. Lee, L. Tsai-Ching, L. de Silva and M. Macy, "Detecting future social unrest in unprocessed Twitter data: Emerging phenomena and big data", Proceedings of 2013 IEEE International Conference on Intelligence and Security Informatics, Seattle, WA, **(2013)**.

[4] Z. Zheng, J. Zhu and L. MR, "Service-generated big data and big data-as-a-service: an overview", Proceedings of 2013 IEEE International Congress on Big Data (Big Data Congress), Santa Clara, **(2013)**.

[5] S. Meng, W. Dou, X. Zhang and J. Chen, "KASR: a key-word-aware service recommendation method on map reduce for big data application", IEEE Transactions on Parallel and Distributed Systems, no. 99, **(2014)**, pp. 1.

[6] J. Cohen, B. Dolan, M. Dunlap, J. Hellerstein and C. Welton, "MAD skills: new analysis practices for big data", PVLDB, vol. 2, no. 2, **(2009)**, pp. 1481-1492.

[7] A. Nascif and K. Sumeyye, "Big data exploration through visual analytics", Proceedings of 2012 IEEE Conference on Visual Analytics Science and Technology, Seattle, WA, **(2012)**.

[8] J. Dittrich and J. Quian´eRuiz, "Efficient big data processing in Hadoop mapreduce", Proceedings of the VLDB Endowment, Istanbul, **(2012)**.

[9] X. Wu, X. Zhu, G. Wu and W. Ding, "Data mining with big data", IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 1, **(2014)**, pp. 97-107.

[10] L. Sun, R. Cheng, D. Cheung and J Cheng, "Mining un-certain data with probabilistic guarantees", Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, **(2010)**.

[11] K. Verlaenen, B. Win and W. Joosen, "Policy analysis using a hybrid semantic reasoning engine", Proceedings of International Workshop on Policies for Distributed Systems and Networks, Bologna, **(2007)**.

[12] W. Lin, Y. Wu, X. Tan and Y. Yu, “An improvement of index method and structure based on R-tree”, Proceedings of International Conference on Computer Science and Software Engineering, Wuhan, **(2008)**.

[13] C. Hernandez-Espinosa, M. Fernandez-Redondo and M. Ortiz-Gomez, “Interval arithmetic inversion: a new rule ex-traction algorithm”, Proceedings of the International Joint Conference on Neural Networks, Singapore. **(2003)**.

[14] Q. Li, W. Wang, D. Xu and X. Wang, “A robust anti-jamming navigation receiver with antenna array and GPS/SINS”, IEEE Communications Letters, vol. 18, no. 3, **(2014)**, pp. 467-470.

[15] J. Bucklew, “Conditional importance sampling estimators”, IEEE Transactions on Information Theory, vol. 51, no. 1, **(2005)**, pp. 143-153.

[16] P. Kinsman and N. Nicolici, “NoC-based FPGA acceleration for monte carlo simulations with applications to SPECT imaging”, IEEE Transactions on Computers, vol. 62, no. 3, **(2013)**, pp. 524-535.

[17] B. Birand, M. Chudnovsky, B. Ries, P. Seymour, G. Zussman and Y. Zwols, “Analyzing the performance of greedy maximal scheduling via local pooling and graph theory”, IEEE/ACM Transactions on Networking, vol. 20, no. 1, **(2012)**, pp. 163-179.

[18] M. Rinehart and M. Dahleh, “The value of side information in shortest path optimization”, IEEE Transactions on Automatic Control, vol. 56, no. 9, **(2011)**, pp. 2038-2049.

[19] J. Cong, A. Kahng and L. Kwok-Shing, “Efficient algorithms for the minimum shortest path Steiner arborescence problem with applications to VLSI physical design”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 17, no. 1, **(1998)**, pp. 24-39.

[20] O. Tataw, T. Rakthanmanon and E. Keogh, “Clustering of symbols using minimal description length”, Proceedings of Document Analysis and Recognition, Washington, **(2013)**.

[21] R. Zuim, J. de Sousa and C. Coelho, “Decision heuristic for Davis Putnam, Loveland and Logemann algorithm satisfiability solving based on cube subtraction”, IET Computers & Digital Techniques, vol. 2, no. 1, **(2008)**, pp. 30-39.

[22] E. Maneva and A. Shokrollahi, “New model for rigorous analysis of LT-codes”, Proceedings of International Symposium on Information Theory, Seattle, WA, **(2006)**.

Author

Zhenhua Huang, received his M.Sc., Ph.D. degree in computer science respectively from Xiamen University and Fudan University China, in 2005 and 2008. He is currently a lecturer in the School of Electronics and Information at Tongji University. His research interests include data warehouse, OLAP application, data mining and knowledge discovery. Since 2004, Dr. Huang has published over 40 papers in various journals and conference proceedings.