Improving user interaction in mobile-cloud database query processing

Chenxiao Wang[1] Florian Helff[1] Farah Kouiss[1] Jason Arenson[1] Le Gruenwald[1] Laurent d'Orazio[2]

[1] School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA
{fhelff, chenxiao, farah.kouiss-1, arensonjt, ggruenwald}@ou.edu

[2] CNRS, UMR 6074, IRISA
Rennes 1 University
Lannion, France
laurent.dorazio@univ-rennes1.fr

ABSTRACT

Choosing an optimal query execution plan to minimize query costs is crucial for the query optimizer, especially in mobile-cloud database systems where there are multiple costs to execute a query plan such as money, time and energy. In order to fulfill different cost objectives for different users, some query optimizers allow users to select the query execution plan from a Pareto Set based on Skyline queries. The users have to select from a potentially large quantity of options and these options are just the values of costs which are not straightforward to the users. This increases the possibility of users to make wrong decisions as well as the time consumption. However, the existing user interaction model during multi-objective query processing is unable to solve this issue. To fill this gap, this paper presents a Normalized Weighted Sum Algorithm (NWSA) and a new user interaction model in multi-objective query processing which introduces the administrators, or super users, to the user interaction process and allows them to preset the Weight Profiles and their logical descriptions that contain the objective preferences for the users before the query is executed. This model allows for super users to set up Weight Profiles that simplify and limit the number of choices a user has to make when working with the system. The user can then choose a Weight Profile that best suits their objective preferences more quickly and accurately. Besides that, we demonstrate a tool to help the super users set up these Weight Profiles and help the users select queries and the logical descriptions of their Weight Profiles for query execution.

INTRODUCTION

In a mobile-cloud database environment, a user issues queries from a mobile device to obtain data from either the cloud database system or a cache on the mobile device. Executing a query incurs three different costs: the monetary cost of query execution on the cloud, the overall query execution time, and the energy consumption on the mobile device where the query might be executed. These three costs constitute the multi-objectives that the query optimizer needs to minimize in order to choose the optimal query execution plan (QEP). Different QEPs are available due to the elasticity of the cloud which considers multiple nodes with different specifications. Meanwhile, different users have different preferences for choosing a suitable QEP for their purposes. Thus, selecting a QEP of skyline queries from the users' perspective becomes an issue. Many early algorithms have been proposed to answer the skyline query [2][3]. Also, some extensive studies made assumptions about the users' purposes without explicit interaction [5][7]. There are some studies that have proposed to use users' feedback as a parameter to guide the skyline QEP selection [4][6][8]. However, none of these works aims at improving the time consumption and accuracy of the users to make such decisions. In our previous work [1], we have presented the NWSA algorithm which allows the user to make a decision for the QEP based on all objectives under a declarative preset guidance. In this paper, in order to simplify the preset process, we present a Super User Interface which allows administrators (or super users) to customize the guidance for their system users by presetting the profiles of the objective preferences (Weight Profiles), and a User Interface which allows users to process queries based on the preset Weight Profiles. The Super User Interface allows the super users to set a limited number of Weight Profiles, which in turn allows the User Interface to present a reduced list of options to the user, making it easier for users to choose a given Weight Profile. The rest of the paper is organized as follows: Section 2 gives a summary of the NWSA algorithm; Section 3 presents the Super User Interface and the User Interface; and Section 4 describes the demo setup.

NORMALIZED WEIGHTED SUM MODEL ALGORITHM

The Weighted Sum Model is an existing optimization strategy which incorporates multiple objectives into its decision. Using a single number called score for each alternative, which includes all objectives, an alternative is rated and can be compared to other alternatives. The score aggregates the different objectives, stressed by individual weights on each objective. Ordering the alternatives by their scores allows the model to choose the best alternative: maximum score for utility functions and minimum score for cost functions. To use the Weighted Sum Model in the context of different dimensions and unit objectives, we presented the NWSA [1], which uses the Weighted Sum Model as a basis but makes major changes to cover the weaknesses. It uses the function in Figure 1 to calculate the score of an alternative (A_i): normalizing n objectives (a_{ij}) to a user-defined maximum of an objective (m_j), eliminating units and resulting in a distribution on a percentage basis. The result of this normalization is then multiplied by a weight (w_j) to individually stress objectives to the preference of a user. The sum of the weighted normalized objectives represents the score of the alternative. These strategies adapt the ideas of a user-based decision.

$$A_i^{WSM-score} = \sum_{j=1}^{n} w_j \frac{a_{ij}}{m_j}$$

Figure 1. Modified Weighted Sum Model Scoring Function

shows the user-interaction models for Skyline Queries / Pareto Set (a) and NWSA (b), respectively. Both algorithms start with the query as their input. NWSA additionally requires the weight profiles to stress the objectives but eliminates the step of a user decision based on the results of the algorithm. This user decision is needed in the Pareto Set Approach since the selection of a single alternative has to be done manually, which pauses the query processing. In the NWSA approach, this decision is made by the algorithm and does not require any interruptions or waiting for additional inputs. The final user decision on an alternative from the result of the Skyline Query remains very complex. Research on the size of a Pareto Set already estimated its size to be $\Theta((\ln n)^{d-1} / (d-1)!)$ for n data objects and d objectives, assuming attribute independence [2].

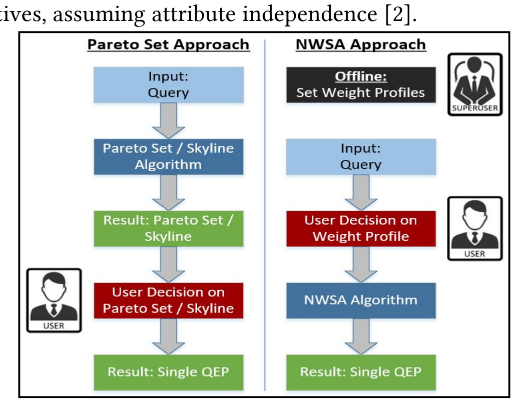

Figure 2. User Interaction Models: Pareto Set Approach and NWSA Approach

SUPER USER AND USER

To eliminate the disadvantage NWSA has in comparison to the Skyline Queries approach in that user preferences have to be known prior to execution, we propose to separate users into two groups: users that make the decision on weights (super users) and users that invoke the execution of a query (users). These preferences of weights are called Weight Profiles. A Weight Profile contains a set of weights that reflect preferences toward the multiple costs. Each Weight Profile with a different cost emphasis is associated with a label, which is an application-based logical description which uses natural language rather than numbers to describe the multiple weights. Several Weight Profiles are set up by the super user before the queries are submitted by the users. There are two aspects of this approach that benefit the users. First, the users do not have to be aware of the details of the Weight Profiles and can select Weight Profiles from their labels. This minimizes the decisions the users have to make. Second, due to how Weight Profiles are created, there will be fewer Weight Profiles than there would be QEPs in Skyline Queries selection or Pareto Set. Fewer options to pick between makes it easier for the user to quickly make the decision correct for them. Making the options declarative reduces the possibility of inaccurate selection and fewer options helps the user make faster decisions. This abstraction to a simpler user interaction reduces the complexity of the user's decision on a weight profile.

Super User Interface

Taking the advantage of super users being able to preset Weight Profiles and making the decision simpler for users still leaves the super user with his/her decision to set weights and label each weight with a description. The super user interface we developed has the purpose of giving an easy tool to super users to set accurate Weight Profiles based on their application requirements for the users. The main feature of this interface is the graphical representation of all possible Weight Profiles in an interactive parallel plot [9]. The super users select a suitable subset and label each Weight Profile with a logical description from all the possible Weight Profiles. The users are only required to select from the labels of this subset. By doing this, the users are only required to select from a small number of labels, with each label representing a suitable QEP preset by the super users. Thus it is less likely for them to make a wrong choice on selecting the QEP. Parallel plots are used to show connections between high-dimensional objectives: a point of an n-dimensional space is represented by n vertices on n parallel dimensions which are connected by a polyline. The position of a vertex in any dimension corresponds to the coordinate of this point in this dimension. Figure 5 shows such a parallel plot with each line representing a Weight Profile (weights on monetary cost, execution time, and energy consumption in axes 2, 4, and 6) and its corresponding cost (axes 1, 3, and 5). The three different costs are calculated based on the historical average cost of a query using this specific Weight Profile, to execute. For the purpose of the demonstration, these values are synthetic. Estimating a QEP is out of scope of this paper, while it is an important perspective in our project. The rectangle on the first axis shows there is a slider on cost axis and this slider applies to each axis. The super users are able to click a line to set the Weight Profile and attach a label to it, as well as use the sliders to set different cost constraints on according cost columns to filter out the QEP whose estimation costs exceed the constraints. For example, the super users can easily use the slider to set the maximum monetary cost according to their operation budget. When each constraint is set, the unavailable lines are filtered and vanish in the interface to indicate that these weights are not suitable for the given query constraints. If the extreme constraints are chosen, it is possible that all the lines vanish which means there are no available weights for the query constraints. The remaining lines represent different profiles which can be saved and attached to a logical description for users to view when selecting Weight Profiles. An example of a Weight Profile would be 0.8 weight on monetary cost, 0.1 weight on execution time, and 0.1 weight on energy with its corresponding costs of $40000, 8100 s, and 220000 mAh. Assuming one million queries, the corresponding average costs per query are $0.04, 0.0081s, and 0.22 mAh when using these weights. If such costs are accepted by the super user, these weights can be saved and labeled with a description of “runs relatively slow but saves money and energy” which indicates that the query optimizer should choose the QEP with a lower monetary and energy cost. In addition, our study shows that it takes less time for an ordinary people to make decision with the help of the logical descriptions when there are more alternatives.

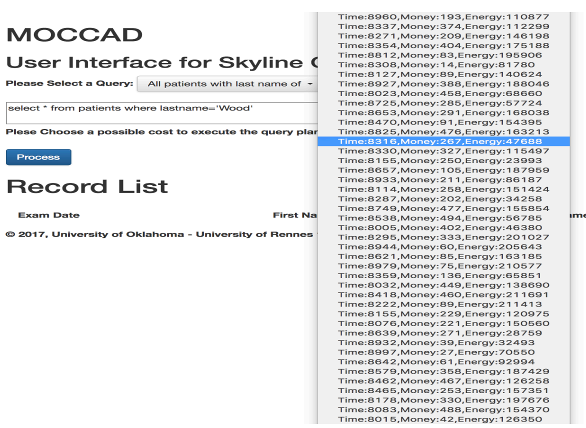

Figure 3. User Interface for Skyline Queries Approach

User Interface

In order to show the advantages of the super user’s presetting of the logical descriptions of the Weight Profiles, we demonstrate a common interface for users.

Figure 3 shows an example of a medical institution application interface used by medical doctors where the Skyline Queries approach is implemented.

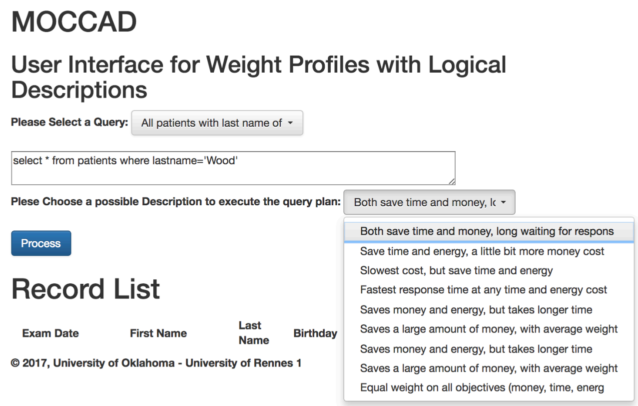

Figure 4. User Interface for Weight Profiles with Logical Descriptions

From Figure 3, we can see that the top part provides different query options for a doctor to select based on his/her purposes. The information that a doctor would like to obtain is displayed after the query has been answered, and the doctor is presented an option to choose how he/she would like to get the information. Notice that, without the Weight Profiles and the logical descriptions, the user has to make a choice from a large number of options produced by the Skyline Queries approach. Thus, this choice can easily be wrong and takes a significant amount of time.

Figure 4 shows a similar interface but with the presets made by the super user. The doctor is only required to choose from a few options and requires no knowledge of weights or the cost of how the data is queried. Only the description of each Weight Profile is visible for the doctor to select based on his/her preferences. This saves time and energy in the doctor’s decision-making process and reduces the chance for the doctor to make a wrong decision in QEP selection. For example, if a doctor is working in the Emergency Room, the doctor selects the Weight Profile with the logical description “Fastest response time at any time and energy cost” in which the query is answered as soon as possible regardless of the monetary cost or how much energy the query will cost the devices. If the doctor is working on his/her medical research and needs a patient’s data, the doctor may choose the Weight Profile with the logical description “Saves money and energy, but takes longer time” to retrieve the data slower but with a lower monetary cost in order to save money. As long as there is a super user that created all the preferences prior to the query execution, and these preferences are declarative to the user, this will significantly simplify the query processing interaction with the users.

DEMO SETUP

The demonstration video [10] shows an example of user interaction in query processing of a medical application.

Data and Queries: Data is a synthetic medical dataset that acts as an example of a dataset with common information of patients.

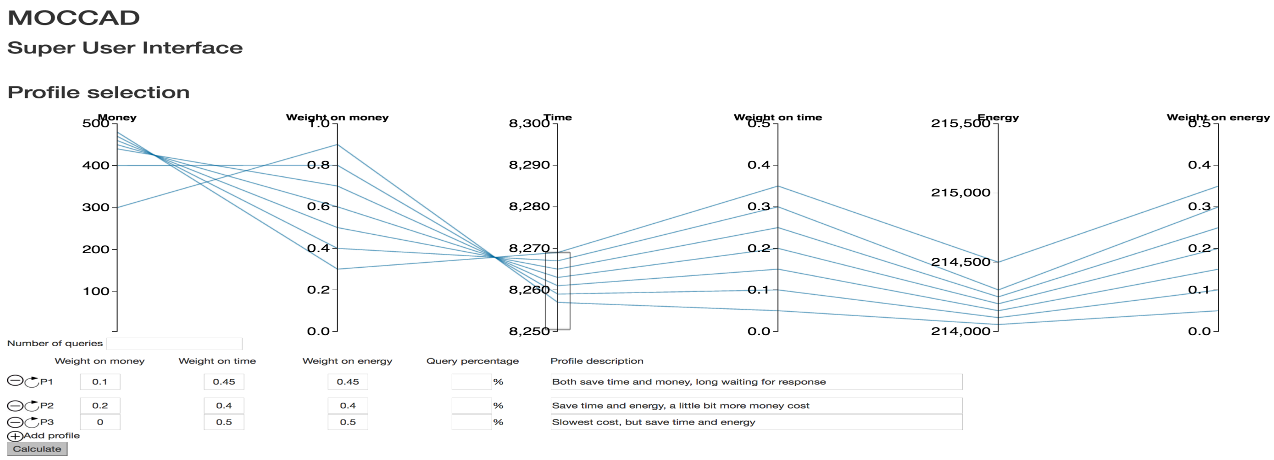

Figure 5. Super User Interface for the super users to select weight profiles and attach logical descriptions.

For queries, we implement queries that imitate a scenario in which a doctor would like to retrieve a patient's information in different situations with respect to time, monetary, and energy costs. Assuming we stored a large amount of medical data on a cloud database system and the data can be retrieved from either desktops or mobile devices, we synthetically simulate different QEPs with different query response times, monetary cost and energy cost under different hardware configurations

Demo. The demo will first show a user interface from the super user's perspective as seen in Figure 5. The super user first presets the Weight Profiles. Before starting, the super user specifies how many queries the system will run. Then the super user uses the slides to filter out the weight profiles for which the query execution plans he/she selects exceed the constraints that the super user might have, and those lines will become invisible. Next, the super user clicks a line to select a weight profile which the super user thinks is suitable for the application. The weight value on the line the super user clicks is displayed. The super user can also specify a percentage of how many queries will use this Weight Profile so that the overall cost of executing the total number of queries can be calculated. After that, the super user can use the "Calculate" button to calculate the estimation of the overall cost of executing these queries with the Weight Profiles selected. When the overall cost is satisfying, a logical description label in natural language is attached to each Weight Profile. Only these labels are visible to the user.

Next the demo will show the user interface from the user's perspective as seen in Figure 4. The user first selects a query from the dropdown menu and then selects a logical description that helps the user make the decision on how the selected query should be executed. After the user clicks on the "Process" button, the selected query will be executed and the query results will be displayed along with the real costs of executing the query. In addition, the demo also shows a similar user interface using the Skyline Queries approach for comparison purposes.

CONCLUSION AND OUTLOOK

This paper presented an algorithm, NWSA, and a user-interaction model that enable user interaction in database query processing with multiple cost objectives. This model was compared with the existing user interaction model of the Skyline/Pareto Set approach. Additionally, this paper demonstrated an interface to allow super users to construct and analyze Weight Profiles needed for users, and an interface to allow users to select a QEP based on the logical description labels of the Weight Profiles. The comparison shows that the user interaction of deciding on a Pareto optimal QEP, which is necessary while using the Skyline Queries approach, can be eliminated by using NWSA. In the future work, we will use three types of user interfaces (1) user interface for Skyline Queries, (2) interface for Weight Profiles and (3) interface for Weight Profile with logical description to develop a user study and invite the users to use these three types of the interface and compare the different time costs each user will spend to make the choice of how the query will be executed. This study will be conducted to show that using a logical description of a Weight Profile substantially increases the accuracy of a user selecting the optimal alternative, and also shortens the time a user needs to select his/her answer.

ACKNOWLEDGEMENT

This work is partially supported by the National Science Foundation Award No. 1349285.

REFERENCES

[1] F. Helff, L. Gruenwald and L. d'Orazio, Weighted Sum Model for Multi-Objective Query Optimization for Mobile-Cloud Database Environments,1st International Workshop on Multi-Engine Data AnaLytics , EDBT/ICDT 2016, Bordeaux, France, 2016.
[2] D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. ACM Trans. Database Syst., 30(1):41–82, 2005.
[3] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with Presorting. In Proceedings of ICDE'03, pages 717–816. IEEE Computer Society, 2003.
[4] W.-T. Balke, U. Güntzer, C. Lofi, User interaction support for incremental refinement of preference-based queries, in: RCIS, 2007.
[5] I Bartolini, Z. Zhang, and D Papadias. Collaborative Filtering with Personalized Skylines. IEEE Trans. on Knowledge and Data Engineering, 23(2), 2011.
[6] D. Kossmann, F. Ramsak, S. Rost, Shooting stars in the sky: an online algorithm for skyline queries, in: VLDB, 2002.
[7] Danupon Nanongkai, Atish Das Sarma, Ashwin Lall, Richard J. Lipton, and Jun Xu. Regret-minimizing representative databases. PVLDB, 3(1-2), 2010.
[8] Jongwuk Lee, Gae-won You, and Seung-won Hwang. Personalized top-k skyline queries in high-dimensional space. Information Systems, 34(1), 2009.
[9] A. Inselberg, Parallel Coordinates: Visual Multidimensional Geometry and Its Applications, Springer,2009
[10] Chenxiao Wang [wang chenxiao] (2011,May 23). Improving user interaction in mobile-cloud database query processing [Video File]. Retrieve from https://youtu.be/UPKLfvWm3gE