Extended Approach for the 2-Phase Query Evaluation (2PE) Scheme

Hyun-Ho Lee, Hong-Kyu Park, Jin-Chul Park, Kil-Hong Joo

1 Yonseong University, South Korea, hhlee@yeonsung.ac.kr,
2 Samsung Electronics Co. Ltd., South Korea, gladiator11@hanmail.net,
3 Hanyang University, South Korea, jin1chul@nate.com
4 Gyeongin National University of Education, South Korea, khjoo@ginue.ac.kr

Abstract. Based on the previously proposed 2PE scheme, this paper proposes a new scheme called a 2-Phase Multiple Query Evaluation (2PME). The proposed scheme extends the 2PE scheme in a way that enables multiple multi-way join continuous queries to be more efficiently processed. It constructs a global execution plan that clusters a common sub-query walk (SQW) shared by more than two queries. Each cluster is executed only one time instead of being executed repeatedly for every related query.

Keywords: data stream, multiple continuous queries, 2PME, multi-way join.

1 Introduction

Ongoing changes in the rapid data streams in a data stream management system (DSMS) may be monitored using a number of pre-registered queries that report results continuously as new data elements of the data streams arrive [1,2]. These queries are called continuous queries. In a previous study, we proposed a 2-phase query evaluation (2PE) scheme for a single multi-way join continuous query. Given a continuous query over multiple data streams, a preprocessing phase detects whether newly arrived stream tuples satisfy the query by identifying the maximum number of candidate tuples that could satisfy it. Subsequently, a evaluation phase identifies the stream tuples that satisfy the query among the candidate tuples specified in the previous phase. In this paper, the proposed 2PE scheme is extended as a way to process multiple continuous queries. It is called a 2-phase multiple query evaluation (2PME) scheme. Given a multi-way join continuous query, the 2PE scheme has defined a query walk, which is a specific order in which all the join predicates of the query are processed. The main issue of the 2PME scheme is how to efficiently share the common sub-query walks among the multiple multi-way join continuous queries, based on the matrix-based approach of 2PE. The proposed scheme shares common sub-query walks by constructing a global execution plan where the sub-query walks that are applicable to more than two continuous queries are clustered. Each cluster is

* Corresponding Author
executed only one time instead of being executed repeatedly for every related query. This enables multiple continuous queries to be processed more efficiently.

2 Preliminaries

The previous study have proposed a new scheme called a 2-phase query evaluation (2PE) for a multi-way joined continuous query. Given a single multi-way join continuous query for multiple data streams, its multiple join predicates can be represented by an undirected graph, called a join graph. In the graph, a distinct node represents each of joined source streams, and an edge between two distinct nodes represents the attribute of the join predicate on the streams corresponding to the nodes. All the join predicates of the query can be arranged in a specific order, called a query walk. In order to process a join predicate efficiently, the proposed approach employs a hash scheme for its join attribute. Given an n-way join query \( Q \) and a hash function \( h \) for the join predicate \( R.a=S.b \) of the query \( Q \) on the streams \( R \) and \( S \), the tuple \( x \) of \( R \) and the tuple \( y \) of \( S \) are buddy tuples if the hashing results of the attributes \( x.a \) and \( y.b \) are the same (i.e. \( h(x.a)=h(y.b) \)). A set of buddy tuples would be a candidate for the matching result of the predicate \( R.a=S.b \). In particular, a set of \( n \) buddy tuples, each of which is a tuple of a single data stream, is a candidate for the final matching result of the query \( Q \). Such a set is called a complete connecting sequence. If a tuple of a single data stream \( S_j \) (\( 1 \leq j \leq n \)) is a part of at least one complete connecting sequence, it is called a connecting tuple of the stream \( S_j \).

In the proposed approach, in order to join each tuple of the current sliding window of each data stream in a multi-way manner, two different types of matrix-based synopses, called window synopsis (\( W \)) and window index synopsis (\( WI \)), are employed. The former maintains the number of tuples assigned to the matrix entry; Meanwhile, the latter identifies each of those tuples that are assigned to the entry by maintaining a list of their addresses. In order to estimate the total number of the result tuples that satisfy a multi-way join query, the proposed approach employs two different types of multiplication operation: matrix multiplication and Element-wise array multiplication.

The window synopsis \( W_i \) is either an \( 1 \times m_k \) vector by the matrix multiplication or an \( m_{k-1} \times m_k \) matrix by the Element-wise array multiplication where \( m_k \) is the number of hash buckets for the whole domain of the join attribute corresponding to the edge \( e_k \).

2PE processes the query \( Q \) symmetrically by the unit of a set of the newly arrived tuples of the specific operand stream; this is called a batch. A series of tasks—inserting, probing, and invalidating—is performed for each batch to produce the result tuples of the query \( Q \). In the preprocessing phase, for each batch of multiple data streams, a pair of a batch synopsis (\( B \)) and a batch index synopsis (\( BI \)) is constructed. The newly arrived tuples of the current batch are inserted into the proposed synopsis \( B \) and \( BI \) along with the synopses \( W \) and \( WI \). The batch synopsis and batch index synopsis are structurally identical to a window synopsis and a window index synopsis, respectively. The evaluation phase generates an accurate set of the final result tuples for the batched synopsis. It is further divided into two steps: Identification and materialization; the identification step refines the entries of all the intermediate result vectors to discard those entries that cannot lead to a complete connecting sequence.
Subsequently, the materialization step produces the result tuples for a batch by only concatenating the successful tuples of each data stream, based on the refined result vectors. It explicitly outputs the final result tuples of the query set \( Q \).

### 3 2-Phase Multiple Query Evaluation Scheme (2PME)

#### 3.1 Sharing common sub-query walks

Given a set of multi-way join queries \( Q = \{ Q_1, Q_2, ..., Q_n \} \), let \( \Psi \) denote the local optimized query walk of each query \( Q_x \). Each query walk \( \Psi \) is generated by the query walk optimization of the 2PE scheme. In order to collectively evaluate the local query walks, the 2PME scheme finds the sharable sub-query walks, known as the sub-query walk (SQW) cluster. Given two query walks, \( \Psi_1 \) and \( \Psi_2 \), the sub-query walk \( \Psi'_1 = < v_1^{p_1} e_1^{p_1} v_1^{p_1+1} e_1^{p_1+1} ... v_1^{q_1}> \) of \( \Psi_1 \) can be shared with the sub-query walk \( \Psi'_2 = < v_2^{p_2} e_2^{p_2} v_2^{p_2+1} e_2^{p_2+1} ... v_2^{q_2}> \) of \( \Psi_2 \) only when the following three conditions are satisfied, since the edges on both ends of the two sub-query walks influence the dimensions of the matrix-based synopses for the sub-query walk.

(i) The two sub-query walks should be same, i.e., \( \Psi'_1 = \Psi'_2 \)

(ii) The edges on the left ends of the two sub-query walks should be same, i.e.,
     \[ e_1^{p_1-1} = e_2^{p_2-1} \text{ or one of the edges is null.} \]

(iii) The edges on the right ends of the two sub-query walks should be same, i.e.,
     \[ e_1^{q_1} = e_2^{q_2} \text{ or one of the edges is null.} \]

After identifying all the sharable SQW clusters, each of the remaining sub-query walks becomes a SQW cluster. As a result, a query walk for a single multi-way join query is represented as a sequence of SQW clusters and the set of the local query walks are integrated into a single global execution plan [4]. The optimal global execution plan is chosen by exhaustive examination of all the possible combinations of the SQW clusters based on the cost of the global execution plan.

#### 3.2 Evaluation of the SQW cluster

Given a global execution plan for a set of multi-way join queries, matrix-based synopses and a query equation for each SQW cluster in the plan are individually constructed. A SQW cluster can be one of the following three types:

i) **Prefix**, when the left edge of the left end node in the cluster \( \hat{C} \) is null;

ii) **Post**, when the right edge of the right end node in the cluster \( \hat{C} \) is null;

iii) **Inner**, otherwise.

Depending on the type of SQW cluster, the matrix-based synopsis for each node in the cluster is constructed in different ways. In a prefix type of SQW cluster \( \hat{C} \), a matrix-based synopsis is constructed for each node in the cluster in the same manner as the 2PE scheme. On the other hand, in an inner type of SQW cluster, if the labels of
two edges on the both end nodes of the cluster are the same, a two-dimensional matrix-based synopsis, whose column and row attribute are same, is constructed for the node, i.e., the synopsis is diagonal while the one-dimensional matrix based synopsis is constructed for the node in the 2PE scheme. Otherwise, the matrix-based synopses are constructed in the same manner as the 2PE scheme. In a post type of SQW cluster, if the labels of two edges on the left end node of the cluster are the same, a two-dimensional matrix-based synopsis, whose column and row attribute are the same, is constructed for the node. Otherwise, the matrix-based synopses are constructed in the same manner as the 2PE scheme.

3.3 Processing a global execution plan

Given a global execution plan for a set of multi-way join queries, let \( C \) denote all the SQW clusters in the query. Let \( Q \) denote a set of the queries with a source stream \( S \), and let \( C \) \(( \subset C \) denote a set of the SQW clusters that should be evaluated for \( Q \). The set \( C \) is divided into two subsets, \( C_B \) and \( C_W \), where \( C_B \) is the set of the SQW clusters with the source stream \( S \), and \( C_W \) is the set of the other clusters, i.e., \( C = C_B \cup C_W \), and \( C_B \cap C_W = \emptyset \). For a new batch of a source stream \( S \), every SQW cluster in \( C \) is individually evaluated in the same manner as the 2PE scheme and their result synopses are constructed. More precisely, matrix-based multiplication operations for the batched (index) stream are evaluated for every SQW cluster in \( C_B \). On the other hand, the matrix-based multiplication operations for the SQW clusters in \( C_W \) remain unchanged. The operations for all the SQW clusters in \( C_B \), and all the SQW clusters in \( C_W \), are evaluated and their results are maintained. Subsequently, since each query in the query set \( Q \) can be modified as a sequence of the result synopses of the SQW clusters in the \( C \), it can be processed by evaluating its modified sequence.

4 Conclusion

This paper proposes a new scheme, called a 2-phase multiple query evaluation (2PME) scheme, to extend the previous 2PE scheme in such a way that multiple continuous queries are efficiently processed. The proposed scheme enables multiple multi-way join queries to be processed more efficiently by sharing their common sub-query walks based on the matrix-based approach of 2PE.
Acknowledgments. This research was supported by Basic Science Research Programs (NRF-2012R1A1A2009170) through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning.

References