A GEOSENSOR FILTER FOR PROCESSING GEOSENSOR QUERIES ON DATA STREAMS

Dong Gyu Lee, Keun Ho Ryu

Database/Bioinformatics Laboratory, Chungbuk National University, Korea
{dglee, khryu}@dblab.chungbuk.ac.kr

ABSTRACT: Pattern matching is increasingly being employed in various researches as health care service, RFID-based system, facility management, and surveillance. Geosensor filter correlates a data stream to match specific patterns in distribution environments. In this paper, we present a geosensor query language to represent efficiently declarative geosensor query. Geosensor operators are proposed to use for fast query processing in terms of spatial and temporal area in distribution environments. We also propose a geosensor filter to match new query predicates into incoming stream predicates. Our filter can reduce the volume of transmission data and save power consumption of sensors. It can be utilized the stream data mining system to process in real-time various data as location, time, and geosensor information in distribution environments.

KEY WORDS: Pattern Matching, Data Stream, Spatiotemporal Queries, Distribution Environment

1. INTRODUCTION

Pattern matching is a new paradigm where events on data streams are matched against specific patterns. Recently, pattern matching on data streams has aroused significant interest in industry due to its wide applicability such as healthcare, facilities monitoring, and surveillance [1, 2, 3, 4]. A lot of Sensor devices have utilized for tracking and monitoring purposes in various areas. These sensors generate massive data stream with various factors and such data are having a volume of events. Applications using these sensor devices require a lot of data streams filtered in real-time. *Publish/subscribe systems* focus mainly on subject or predicate-based filters over individual streams. *Stream processing systems* are not optimized for complex stream processing whereas data stream processing systems nowadays developed do not support fast execution.

In this paper, we present a geosensor filter to match query predicates into incoming stream predicates in distribution environments. In particular, we explain three challenges that arise in data stream processing on data streams. First, massive data streams generated by wide deployments of sensors can be obtained per second or higher. Geosensor filter have to be keep up with these massive data streams. Second, monitoring applications sometimes apply a sliding window (e.g., within last 3 hours) to a sequence of streams of interest. A simple event detection responses only answers of a query. However, relevant events with all results increase significantly the execution time. Third, our study is performed in distribution environments as Figure 1. A user can obtain data streams in their local area. Existed studies have worked in server-client system or ad-hoc networks. Previous work filters out incoming raw data from sensors in a server side as end-device of ad-hoc networks. It causes heavy network traffics and power consumption due to a volume of data transferred from geosensors.

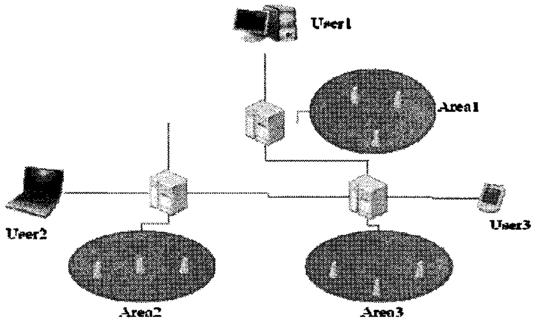

Figure 1. Geosensor Distribution Environment

In distribution environment, we suppose that User1 want to know the dust density for last 3 hours in Area2. We can only obtain data need from specific sensors of narrow area. So our geosensor filter can reduce network traffics and save power consumption in distribution environment.

In the remainder of the paper, we not only describe geosensor query language and geosensor operators, but also represent proposed geosensor filter. Finally, we make a conclusion as summarization.

2. GEOSENSOR QUERY LANGUAGE

In this section, we describe geosensor query language with an example and explain geosensor operators for geosensor query processing.

2.1 Overall Structure

Our geosensor query language is a declarative language that includes spatial and temporal conditions. It can be used to specify that individual events are filtered. This declarative language is consisted of five clauses as

CONTEXT, QUALIFICATION, FROM, LOCATION, and DURATION. Following is overall structure of geosensor query:

CONTEXT <context definition>
QUALIFICATION <context condition>
FROM <target>
LOCATION <spatial expression>
DURATION <temporal expression>

In CONTEXT clause, contexts of users are defined. QUALIFICATION clause expresses context condition of users like air pollution, fire disaster, flood disaster, and weather in specific area. Targets like sensors or users in specific area are decided in FROM clause. LOCATION and DURATION clauses mean location of targets and a time duration for required data. Geosensor operators are used in Individual clause that query processing is completed efficiently.

2.2 Example of Proposed language

We now explain an example for a usage of proposed geosensor query language. Query 1 defines air pollution with specific conditions. Conditions of air pollution are consisted of dust density, CO2, and temperature elements. FROM clause expresses targets for gathering required data as 'sensors'. LOCATION clause means that the sensors are 'all' in only 'Area1'. DURATION clause means that the sensors are gathered for only 3 hours. Namely, Query 1 is to inform data as dust, co2, temperature gathering for 3 hours from whole sensors is located in Area1. QUALIFICATION clause is made up three combinations of predicates.

Atomic predicate: comparison of the variable and constant such as {<, <=, =, !=, >=, >}.
Conjunctive predicate: combination of atomic predicates such as {(k<=11) and (l>=17)}.
Disjunctive predicate: combination of conjunction predicates such as {((k<=11) and (l>=17)) or ((k>=13) and (l>=20))}.

Query 1:
CONTEXT air pollution [dust D, CO2 C, temperature T]
QUALIFICATION D>100 or C>1 or T>28
FROM sensors
LOCATION Region [Area1, all]
DURATION Interval [3hours]

Proposed language is efficient to reduce network traffic and save sensor power and process data stream using geosensor operators as Region, Interval and so on.

2.3 Geosensor operators

Geosensor operators can process geosensor queries efficiently in distribution environments. geosensor operation returns true/false values against targets based on temporal and spatial factor. Typical spatial operators are direction, distance, and length, and temporal operators are overlap, and extend.

Table 1. Operators for Geosensor Query Processing

Operator	Description
Min	Return a minimum value
Max	Return a maximum value
Region	Examine to be present target(s) in specific region
Interval	An unanchored contiguous portion of the time line
Period	An anchored duration of the time line
Instant	An anchored location on the time line

Table 1 shows geosensor operators used in distribution environments. Min and Max operator return a minimum and maximum value for specific time duration. Region defines to examine to be present targets in specific region. Interval operator is used to maintain an unanchored contiguous portion of the time line. Period operator is defined to maintain an anchored duration of the time line. Instant operator is an anchored location on the time line. Geosensor filter refines incoming data stream from sensors for specific duration using interval, period and instant operator.

3. GEOSENSOR FILTER

Geosensor filter is consisted of 2 components as Boundary Check, and Rectangle match as figure 2. Proposed filter is located in base stations gathering data streams in local areas. In base stations, data stream is obtained based on a disjunctive predicate for DURATION in local area that satisfies LOCATION clause. Required data is only transferred to users. It can be reduce network traffic and save sensor power consumption.

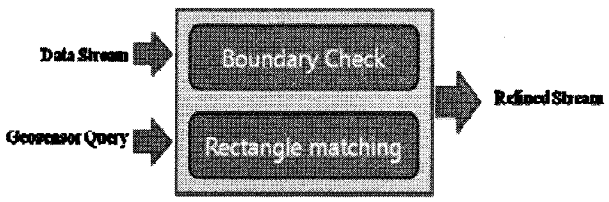

Figure 2. Geosensor Filter

3.1 Boundary Check

Figure 3 illustrates boundary check algorithm which it checks intersection points of two boundaries(e.g., query predicate and stream predicate). It has 5 flags in terms of intersection between query and stream boundary. When two boundaries are no interaction, the flag sets 0. When two boundaries are the same, the flag sets 1. When a stream boundary contains a query boundary completely, the flag sets 2. When intersecting two boundaries, flag sets 3. When a query boundary contains a stream boundary completely, flag is 4. The result of boundary check operation is transferred to Rectangle match component.

Procedure Boundary Check(BC)

Input: query boundary(QB) and stream boundary(SB).
Output: compatibility status of QB and SB

Begin
Check a boundary of query and stream boundary
If no intersection of QB and SB **then** flag is zero
Elseif QB and SB are the same **then** flag is one
Elseif QB is totally contained by SB **then** flag is two
Elseif QB and SB are intersected **then** flag is three
Else SB is inside in QB **then** flag is four
End

Figure 3. Boundary Check Operation

3.2 Rectangle match

Rectangle matching operation performs two matching operation as Simple matching and Partial matching. Figure 4 illustrates to perform Simple and Partial Matching Operation according to a value of flag transferred from Boundary Check Operation. It checks a value of flag computed from Boundary Check Operation. If the flag is zero, there is no matching. If the flag is one or two, Simple Matching is performed. If the flag is three or four, Partial Matching is performed.

Procedure Rectangle Matching(RM)

Input: a constant value of flag from BC
Output: data streams matched to query boundary

Begin
Check a flag transferred from BC
If flag is one or two **then** Simple Matching
Elseif flag is three or four **then** Partial Matching
Else no matching
End

Figure 4. Rectangle Matching Operation

Figure 5 shows two query rectangles are overlapped with four stream rectangles. Query1 is matched to stream rectangle at three times for same time duration and then return the result. However, Query2 is matched to stream rectangle at only one time. Query1 is matched by Partial Matching and Query2 is matched by Simple Matching. Also, a query with long duration is performed by Partial Matching.

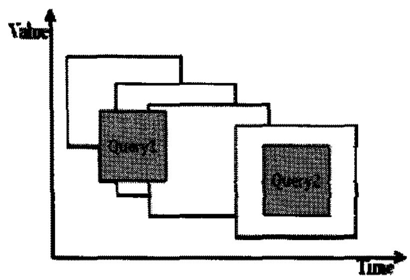

Figure 5. Simple and Partial Matching

4. SUMMARY

We proposed a query language and a geosensor filter to match queries and data streams efficiently in distribution environments. Proposed query language is efficient to express declarative geosensor query. Geosensor operators can be computed efficiently according to spatial and temporal factor in distribution environments. Our filter can reduce a volume of transmission data and save power consumption of sensors.

Currently, we are extending the geosensor operators and implementing the proposed filter. The proposed filter architecture will be also evaluated to be efficient to match geosensor queries into data streams.

References:

[1] Babu, S., Motwani, R., Munagala, K., Nishizawa, I., and Widom, J., 2004. Adaptive ordering of pipelined stream filters. The 2004 ACM SIGMOD international conference on Management of data, Paris, France, pp 407-418.

[2] Dittrich, J.P., Fischer, P.M., and Kossmann, D., 2005. AGILE: adaptive indexing for context-aware information filters. The 2005 ACM SIGMOD international conference on Management of data, Baltimore, Maryland, USA, pp 215-226.

[3] Wu, E., Diao, Y., and Rizvi, S., 2006. High-Performance complex event processing over streams. The 2006 ACM SIGMOD international conference on Management of data, Chicago, IL, USA, pp. 407-418.

[4] Agrawal, J., Diao, Y., Gyllstrom, D., and Immerman, N., 2008. Efficient pattern matching over event streams. The 2008 ACM SIGMOD international conference on Management of data, Vancouver, Canada, pp 147-160.

Acknowledgements

This research was supported by a grant(#07-KLSG-C05) from Cutting-edge Urban Development – Korean Land Spatialization Research Project funded by Ministry of Construction & Transportation of Korean government.