Efficient Processing of Sensor Network Data using Object-Oriented Databases

Kyung-Chang Kim¹, and Choung-Seok Kim²
¹Department of Computer Engineering, Hongik University, Seoul, South Korea
²Department of Information Technology, Silla University, Pusan, South Korea

Abstract - Many wireless sensor network (WSN) applications require join of sensor data belonging to various sensor nodes. For join processing, it is important to minimize the communication cost since it is the main consumer of battery power. Most join techniques for sensor data assume that the sensor data are either stored in OS files or in relational databases. In this paper, we introduce a join technique for sensor networks based on column-oriented databases. A column-oriented database store table data column-wise rather than row-wise as in traditional relational databases. The proposed algorithm is energy-efficient since, unlike relational databases, only relevant columns are shipped to the join region for final join processing.

Keywords: Wireless sensor network, sensor data, join technique, column-oriented database, communication cost.

1 Introduction

Many sensor network applications require correlation of sensor readings scattered among sensor nodes. For example, in an object tracking system, one may be interested in objects that travelled from one designated region to another designated region to monitor the traffic volume and speed of particular objects. The sensor network can be modeled as a distributed database and the sensor readings are collected and processed using queries. A join is an important operation in finding the correlation of sensor readings.

One of the most important performance criteria in processing a join operation for sensor networks is to minimize the total communication cost. A total communication cost is the total data transfer between neighboring sensor nodes. Minimizing the communication cost is important because each sensor node has limited battery power and data communication is the main consumer of battery energy.

In this paper, we propose a novel join technique for wireless sensor networks to minimize the total communication cost and improve performance. Our approach is based on a column-oriented databases rather than relational databases to store sensor data. Recent years have seen an increased attention and research work on column-oriented databases. A column-oriented database store data in column order (i.e. column-wise) and not in row order as in traditional relational databases. They are more I/O efficient for read-only queries since they only access those columns (or attributes) required by the query. The read-only queries are common in workloads and applications found in data analysis, semantic web and sensor networks. The paper is organized as follows. Related works are mentioned in Section 2. Section 3 discusses the proposed algorithm. Conclusion is made in Section 4.

2 Related Works

Query optimization techniques for column-oriented databases were introduced in the literature. Materialization strategies, both early and late, are important in column-oriented databases since tuple reconstruction is required to produce and display query result using conventional ODBC interface [1]. Our earlier join algorithm for sensor networks based on column-oriented database was proposed in [2]. The algorithm is based on an early materialized strategy in column-oriented database. The invisible join [3] is another research result to improve star schema queries using a column-oriented database. Several energy-efficient query processing algorithms were proposed for wireless sensor networks [4-5].

3 Proposed Algorithm

In this paper, we assume that the sensor data are stored in a column-oriented database rather than a relational database. There are two materialization strategies, early and late, in column-oriented databases [2]. Materialization, also known as tuple stitching or tuple construction, is a process of combining single-column projections into wider tuples. The reason for materialization in column-oriented databases is because it needs to output row-style tuples to support standards-compliant relational database interface such as ODBC and JDBC. In early materialization strategy, each column is added to the intermediate query result to form tuples if the column is needed. In late materialization strategy, the accessed columns do not form tuples until after some part of the query plan has been processed. We propose an in-network join technique based on early and late materialized strategy for column-oriented databases.

We assume that the join query involves the join of sensor data in R region and S region of the sensor network. In addition, the actual join is performed in the join region since no single node can perform join due to resource limitations.
The data in R (S) region is stored in R (S) table in column-order. The proposed join algorithm executes in three phases, namely the selection phase, the join phase, and the result phase. In the selection phase, the semi-join columns of R and S is performed to determine the qualified columns and column values for the given query. The result of the semi-join is used to create bitmaps to be shipped back to region R and region S. A Bitmap(R) contains one bit for every tuple in R table. That bit is set to 1 if it is in the semi-join result. The qualified column values in R and S are shipped to the join region for the actual join. In the join phase, the qualified column values of R and S are joined using either a nested-loop or a sort-merge join algorithm. In the result phase, the result of join is sent to the query sink as the query result.

In the late materialized strategy, the qualified column values of R and S are stitched together to construct tuples in the result phase. In the early materialized strategy, the columns in R and S participating in the query result are stitched to form tuples before shipping to the join region for final join. The difference between the early and late materialized strategy is the timing of the tuple stitching. In late materialized strategy, it is performed in the result phase while in the early materialized strategy it is performed in the selection phase.

In this paper, we also introduce a hybrid strategy to reduce the communication cost. Both the early and late materialized strategy performs the semi-join in the semi-join region which is not the join region. In addition, the actual join is also performed in the join region for the reduced data. In the hybrid case, no distinction is made between the semi-join region and the final join region. Both the semi-join and the final join are performed in the join region. Hence, no semi-join region is required.

4 Conclusions

We introduced a join technique that can be used in wireless sensor networks to reduce communication cost during sensor query processing. Communication cost is the main query performance criteria in sensor networks. Our algorithm is based on column-oriented databases. To the best of our knowledge, no other research results for sensor networks based on column-oriented databases were published in the literature. The advantage of using column-oriented database is that only relevant columns of qualified records, not the whole records, need to be shipped to the join region for final join reducing the amount of data shipped. Our algorithm is based on both the early and late materialized strategies. We also introduced a hybrid approach in which the semi-join and the join processing both occur in the same region of the sensor network. It is easy to observe that the communication cost is reduced during join processing based on column-oriented databases compared to data storage based on relational databases.

5 Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number 2012-0007012)

6 References


