International Journal of Engineering & Technology, 7 (4.19) (2018) 1060-1065

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Scalable Execution of KNN Queries using Data Parallelism Approach

Kalpana V. Metre[1]*, M. U. Kharat[2]

[1,2] *Department of Computer Engineering, MET's Institute of Engineering, Nashik, India.*
**Corresponding author E-mail: kvmetre@gmail.com*

Abstract

In recent years, real-time data-oriented applications such as sensor networks, telecommunications data management, network monitoring are required to process various continuous queries on unbounded data streams. A lot of work has been done to deal with the computational complications in constant processing of continuous queries on unbounded, continuous data stream. The K-nearest neighbor algorithm (KNN) is a well-known learning method used in a wide range of problem-solving domains e.g., network monitoring, data mining, and image processing etc. The efficient and scalable processing of multiple continuous queries on dynamic data items requires query indexing and data indexing. Query processing algorithms used on static databases are not well suited to handle dynamic continuous queries over high dimensional data sets. It is better to build the index for queries which is finite rather than to build the index for data which is infinite. A divide-and-conquer approach is used for indexing and searching for K-nearest neighbors. The approach significantly will reduce the space complexity and will scale well with the increasing data size. The hybrid indexing approach using grid and a K-dimensional tree will reduce the space cost as well searching cost. The data parallelism will provide scalability of continuous queries over high-volume streams.

Keywords: Data stream; Grid indexing; KNN; R-tree; Scalability

1. Introduction

In recent years, continuous queries process real-time data streams in applications such as sensor networks, network monitoring, stock market etc. Unlike regular queries, a continuous query is evaluated regularly over a period of time. Traditional data processing algorithms are not well suited to handle various continuous queries over data streams. The previous researchers used either R-tree based indexing for data which is continuous and infinite. The maintenance and updating cost of those disk-based indexing is more than the query response time. Some of the earlier methods are not capable of handling the multi-dimensional data stream. If the response time of continuous query processing is larger than the rate of the input stream tuples, the delays are accumulated and it prevents the system from keeping up with the input stream rate. In many real-time applications, continuous KNN queries are processed on high dimensional dynamic data items. The K-nearest neighbor algorithm (KNN) is a well-known learning method or statistical search used in a numerous applications such as sensor networks, data mining and image processing etc. The advantages of KNN algorithm include that it is fairly simple to implement and it is well suited for multi-modal applications. The KNN search implementations have high computational costs, especially when used with a large amount of high dimensional data. Many KNN implementations degrade in performance as the data becomes high dimensional. Another drawback of KNN concerns its significant memory requirements, especially for indexing massive high dimensional data. The indexing techniques for the efficient and scalable processing of multiple continuous queries on dynamic data items include query indexing and data indexing. With dynamic data, frequent updates to the index on data are inevitable. The data indexing becomes expensive as the data is not persistent, but it is in large volume. So, a new approach is suggested to build the index for queries which is finite rather than to build the index for data which is infinite. The rate of change of queries is low compared to that of data stream, so the query indexing methods can reduce the index maintenance cost [1]. This finite index on queries can be accommodated in memory which results in efficient execution of queries avoiding memory access frequently. The techniques used for building the index on queries include mostly grid indexing and tree-based indexing. The R-tree-based methods are suitable for location-based applications, but due to overlapping structure of tree, the methods discussed in [2] suffer from high maintenance cost. This approach gives good performance only for small numbers of queries.

The grid indexing divides [3][4][5] the d-dimensional indexing space into equal-sized cells and these cells are used for query indexing. It can give lower maintenance cost for dynamic queries and better performance than the other indexing such as tree-based methods [5]. The continuous queries can be static as well as dynamic. In applications such as location-based services, we need dynamic queries. The query indexing approaches need to tackle changing continuous queries [7] and KNN queries [10]. H. Wang et al. proposed an infrastructure MOVNET which combined R-tree structure to store the road network connectivity information and grid index to efficiently process moving object position updates. The limitation is that it can handle only stationary network [8]. W. Choi et al. introduced a new indexing structure that re-

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.