[1]시공간 질의 클러스터링: 데이터 큐빙 기법

심상예, 백성하, 배해영
인하대학교 컴퓨터정보공학과
e-mail : airchendb@gmail.com, shbaek@dblab.inha.ac.kr, hybae@inha.ac.kr

[1]Spatio-temporal Query Clustering: A Data Cubing Approach

Xiangrui Chen, Sung-Ha Baek, Hae-Young Bae
Dept. of Computer and Information Engineering, Inha University

요 약(Abstract)

Multi-query optimization (MQO) is a critical research issue in the real-time data stream management system (DSMS). We propose to address this problem in the ubiquitous GIS (u-GIS) environment, focusing on grouping 'similar' spatio-temporal queries incrementally into N clusters so that they can be processed virtually as N queries. By minimizing N, the overlaps in the data requirements of the raw queries can be avoided, which implies the reducing of the total disk I/O cost. In this paper, we define the spatio-temporal query clustering problem and give a data cubing approach (Q-cube), which is expected to be implemented in the cloud computing paradigm.

1. Introduction

Ubiquitous computing plays a more and more important role in our daily life. There are two typical platforms that support ubiquitous service, *Web* and *Sensor Networks*. For the latter one, advances in sensor technology and deployment strategies are making Geosensor Network (GSN) a primary platform. GSN are defined as the specialized applications of wireless sensor network (WSN) technology in geographic space, which detects, monitors, and tracks environmental phenomena and processes [1].

Within GSN paradigm, data about the real world is collected from the widely deployed geosensors continuously in the form of data stream [2]. Intensive research work has been done on data stream processing and many prototype data stream management systems (DSMSs) are demonstrated, e.g. STREAM, TelegraphCQ, etc. However, due to the location dependent characteristic of geosensor, specialized spatio-temporal (ST) data stream processing issues should be considered. PLACE is the first ST-DSMS [3].

Given *M* raw queries, multi-query optimization (MQO) is a long-term research topic, which is also considered in ST-DSMS [4]. In this paper, we propose to address this problem by grouping 'similar' ST queries incrementally into N clusters so that they can be processed virtually as N queries ($N<<M$). By minimizing N, the overlaps in the data requirements of the raw queries can be avoided, which implies the reducing of the total disk I/O times from *M* to *N*.

Because of the limit of space, we will only attempt to define the spatio-temporal query clustering problem and introduce a data cubing approach in the following sections.

2. Spatio-temporal Query Clustering

Query clustering was first defined in [5], which involves determining globally optimal execution strategies for a set of queries. It addresses the problem from the perspective of overlaps in data requirements and models the batched operations using a set-partitioning approach. Researchers generalize this idea into different applications [6-10], e.g. XML continuous queries [7], continuous quantile queries [8], Top-N selection queries [9]. Although region clustering is researched in [9], it does not consider the case when queries come in as a stream. Furthermore, no generalized clustering method for continuous ST queries was proposed.

DBMSs are still I/O bound and therefore it is critical how many disk accesses is necessary and how good caching is adopted. Likewise, the spatio-temporal query processing is mainly challenged by accessing spatial data in several aspects, e.g. maps, even with multi-dimensional indices like R-tree and its variants. First, it is impossible to load all data into memory entirely at one time. Second, when large amounts of queries are registered, the disk I/O frequency will be increased linearly if each query is processed individually. Third, there are lots of random accesses to pages regarding spatio-temporal processing, which are much slower than reading pages in sequential order. This is because the current operating systems and hard disks provide a pre-fetching strategy (caching). Besides, the random positioning of the read/write head of a hard disk composes the most of the response time. Facing this 'scalability' problem, it will be more cost effective to run 'similar' queries collectively than individually. Especially in the query stream environment, where large amounts of queries come in as a stream, stable processing ability can be guaranteed by query clustering.

Definition 1 (Spatio-temporal Query Clustering). Given a set of spatio-temporal queries Q_i ($i>0$), 'similar' queries are grouped together as clusters QC_j ($j>0$), where queries in each cluster QC_j are processed independent of the others.

[1] This research was supported by a grant (07KLSGC05) from Cutting-edge Urban Development - Korean Land Spatialization Research Project funded by Ministry of Construction & Transportation of Korean government.

To address the problem defined above, several research issues should be considered carefully. For example, how to represent the ST queries uniformly? How to compute the query similarity and incrementally group new coming query stream into limited amounts of clusters? How to execute each query cluster and to what extent can it improve the query efficiency and scalability?

3. A Data Cubing Approach

It is essential that the underlying theme of the works in spatio-temporal query processing is multi-dimensional query processing, where both the geographic data and non-spatial information need to be aggregated. Motivated by the attempts of integrating GIS and Online Analytical Processing (OLAP) [11], we propose to do ST query clustering through data cubing approach.

The original data cube proposed by J. Gray supports complex Group-bys well, which made it become one of the most popular commercialized technologies in DB area [12]. However, it considers RDBMS and needs to be improved to support more new data type, e.g. data stream [13], XML like unstructured data, uncertain data, etc. In our research, we will follow the diagram and construct a Query-cube (Q-cube).

In our work, ST queries are represented by a feature table, where dimensions of information uniformly summarize the general ST query properties and features. For different dimensions, concept hierarchies are abstracted according to the archived spatial data. Obviously, it is then easy to construct a data cube to represent and cache the concurrent registered queries, which is named Q-cube. Based on this formal query model (Q-cube), clustering algorithms relying on the dynamically maintained Q-cube will be researched. Moreover, to support specialized personal ubiquitous service in mobile environment, several embedded query processing methods should be given against the subspace inside Q-cube (i.e. cuboids). In addition, the query execution strategy against continuous data stream will be considered, including both the time-based and change-based queries.

4. Summary

In real applications, it is more cost effective to provide the ubiquitous services based on distributed paradigm, e.g. Cloud Computing (CC). Being the main-stream technique for CC, MapReduce [14] is proved to be more and more effective in various areas. Previous work has proved the feasibility of our idea [15-20]. We will attempt to implement relevant methods following this programming paradigm. The DSMS developed at Database Lab. of Inha University will be adopted for ST query processing.

In this paper, we review the issue of query clustering and define the specified spatio-temporal query clustering problem. A data cubing approach is introduced briefly. More detailed work will be published in the future papers.

참고문헌(Reference)

[1] S. Nittel, “A Survey of Geosensor Networks: Advances in Dynamic Environmental Monitoring.” Sensors Journal, Vol.9, No.7, pp.5664-5678, 2009.

[2] B. Babcock, S. Babu, M. Datar, R. Motwani and J. Widom, “Models and Issues in Data Stream Systems.” PODS, Madison, Wisconsin, USA, pp.1-16, 2002.

[3] M. Mokbel and W. Aref, “PLACE: A Scalable Location-aware Database Server for Spatio-temporal Data Streams.” Data Eng. Bull., Vol.28, No.3, pp.3-10, 2005.

[4] M. Mokbel and W. Aref, “SOLE: Scalable Online Execution of Continuous Queries on Spatio-temporal Data Streams.”VLDB J., Vol.17,No.5, pp.971-995, 2008.

[5] R. Gopal and R. Ramesh, “The Query Clustering Problem: A Set Partitioning Approach.” IEEE TKDE, Vol.7, No.6, pp.885-899, 1995.

[6] A. Ghosh, J. Parikh, V. Sengar and J. Haritsa, “Plan Selection based on Query Clustering.” VLDB, Hong Kong, China, pp.179-190, 2002.

[7] J. Chen, D. DeWitt, F. Tian and Y. Wang, “NiagaraCQ: A Scalable Continuous Query System for Internet Databases.” SIGMOD, Texas, USA, pp.179-190, 2002.

[8] X. Lin, J. Xu, Q. Zhang, H. Lu, J. Yu, X. Zhou and Y. Yuan, “Approximate Processing of Massive Continuous Quantile Queries over High-Speed Data Streams.” IEEE TKDE, Vol.18, No.5, pp.683-698, 2006.

[9] L. Zhu, W. Meng, W. Yang and C. Liu, “Region Clustering based Evaluation of Multiple Top-N Selection Queries.” Data & Knowledge Engineering Journal, Vol.64, No.2, pp.439-461, 2008.

[10] S. Xiang, H. Lim and K. Tan, “Impact of Multi-query Optimization in Sensor Networks.” DMSN, Seoul, South Korea, pp.7-12, 2006.

[11] L. Gomez, S. Haesevoets, B. Kuijpers and A. Vaisman, “Spatial Aggregation: Data Model and Implementation.” Information System J., Vol.34, No.6, pp.551-576, 2009.

[12] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow and H. Pirahesh, “Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab and Sub-Totals.” DMKD J., Vol.1, No.1, pp.29-53, 1997.

[13] J. Han, Y. Chen, G. Dong, J. Pei, B.W. Wah, J. Wang and D. Cai, “Stream Cube: An Architecture for Multi-Dimensional Analysis of Data Streams.” Distributed and Parallel Databases J., Vol.18, No.2, pp.173-197, 2005.

[14] J. Dean and S. Ghemawat, “MapReduce : Simplified Data Processing on Large Clusters.” OSDI, USENIX Association, San Francisco, CA, USA, pp.137-150, 2004.

[15] A. Sarje and S. Aluru, “A MapReduce Style Framework for Trees.” Technical Report, ISU, USA, 2009.

[16] Q. Chen, L. Wang and Z. Shang, “MRGIS: A MapReduce-Enabled High Performance Workflow System for GIS.” SWBES, USA, pp.646-651, 2008.

[17] A. Cary, Z. Sun, V. Hristidis and N. Rishe, “Experiences on Processing Spatial Data with MapReduce.” SSDBM, New Orleans, LA, USA, pp.302-319, 2009.

[18] B. Yang, Q. Ma, W. Qian and A. Zhou, “TRUSTER: TRajectory Data Processing on ClUSTERs.” DASFAA, Brisbane, Queensland, Australia, pp.768-771, 2009.

[19] J. You, J. Xi, P. Zhang and H. Chen, “A Parallel Algorithm for Closed Cube Computation.” IEEE/ACIS, Istanbul, Turkey, pp.95-99, 2008.

[20] K. Sergey and K. Yury, “A Parallel Algorithm for Closed Cube Computation.” DBKDA , Cancun, Mexico, pp.62-67, 2009.