International Research Journal of Computer Science (IRJCS)
Issue 08, Volume 4 (August 2017)

ISSN: 2393-9842
www.irjcs.com

A REVIEW OF CONTINUOUS QUERY (CQ) APPLICATIONS

D.C.Igweze
Department of Computer Science,
University of Port Harcourt, Nigeria
dcigweze@yahoo.com,

Prof. E.O. Nwachukwu
Department of Computer Science,
University of Port Harcourt, Nigeria
enoch.nwachukwu@uniport.edu.ng

Manuscript History
Number: **IRJCS/RS/Vol.04/Issue08/AUCS10086**
DOI: 10.26562/IRJCS.2017.AUCS10086
Received: 09, August 2017
Final Correction: 14, August 2017
Final Accepted: 19, August 2017
Published: August 2017
Citation:D.C, Igweze. & E.O., Nwachukwu. (2017). A Review of continuous Query (CQ) Applications IRJCS:: International Research Journal of Computer Science, IV, 30-32. doi: 10.26562/IRJCS.2017.AUCS10086
Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India

Abstract: In many recent applications, data may take the form of continuous data streams, rather than finite stored data sets. Several aspects of data management need to be re- considered in the presence of data streams, offering a new research direction for the database community**.** Consequently, research has moved from the traditional database management system (DBMS) to the data stream management system (DSMS) which is powered by continuous queries. In this paper, continuous query (CQ) is introduced to motivate our discussion of its application areas.

Keywords: Applications, Continuous queries, Data streams, finite, Database.

I. INTRODUCTION TO CONTINUOUS QUERY

A query when run against a data region yields a result set. This result set is essentially a static snapshot of the query result based on the existing data in the region. If the data changes after the query was executed, then the caller has to re-execute the query at periodic intervals to obtain the new changes. This is neither scalable, and not a particularly efficient way to keep the result set up to date for the caller. A Continuous Query solves this problem by allowing the caller to register the query with the server.

With the Continuous Query, the query returns a result set to the caller but continues to evaluate any data change to the region, if it qualifies as a result for the Continuous Query it sends the changes to the caller for inclusion in its original result set, thus ensuring the result set to be always in sync with the contents of the data region. For this reason, Continuous Queries are also referred as Active Queries. Continuous queries are persistent queries that allow users to receive new results when they become available [4]. Continuous query systems can transform a passive web and databases into an active environment; Continuous queries ([5]; [6]) allow users to obtain new results from a database without having to issue the same query repeatedly.

Continuous queries are especially useful in an environment like the Internet, comprised of large amounts of frequently changing information. In order to handle a large number of users with diverse interests, a continuous query system must be capable of supporting a large number of triggers expressed as complex queries against web resident data sets.

Some of the key features of Continuous queries (CQs) are:

- Standard query syntax and semantics.
- Complete integration with the client/server architecture:
- CQ functionality is fully integrated with the client/server architecture. As a result, CQs are highly available; provide transparent failover capabilities, and durable CQs can be registered by clients that are configured to be durable.
- Active query execution—once initialized, the queries only operate on new events instead of on the entire region data set. Events that change the query result are sent to the client immediately.

II. CLASSIFICATION OF CONTINUOUS QUERY

Continuous queries can be classified into two categories depending on the criteria used to trigger their execution: change-based and timer-based continuous queries. A query will become effective at the start-time. The Time-interval indicates how often the query is to be executed. A query is timer-based if its time-interval is not zero; otherwise, it is change-based [5].

Change-based continuous queries are fired as soon as new relevant data becomes available. Two types of events detection can trigger continuous queries. They are data-source change events and timer events. Timer-based continuous queries are executed only at time intervals specified by the submitting user. For instance, day traders would probably want to know the desired price information immediately, while longer-term investors may be satisfied being notified every hour [3]. Although change-based continuous queries obviously provide better response time, they waste system resources when instantaneous answers are not really required. Since timer-based continuous queries can be supported more efficiently, query systems that support timer-based continuous queries should be much more scalable. However, since users can specify various overlapping time intervals for their continuous queries, grouping timer-based queries is much more difficult than grouping purely change-based queries [4].

III. APPLICATION AREAS OF CQ

Continuous queries are enabling data structures that have continued to power many business, commercial, and technologically driven applications. Most location-based service (LBS) applications, online transaction processing (OLTP) applications, alarm systems, etc. have continuous queries as part of their programs building blocks.

Online Transaction Processing (OLTP) Applications: OLTP application like bank account transactions can be configured to send sms (short message service) alerts to the account holder once there is a change in the account balance or database. Also, bank statements of accounts can be sent to account holders at monthly basis. These alerts systems are made possible because of the continuous query component of the system which is set to trigger once there is a change in the database or at specified time intervals i.e., changed-based and timer-based queries respectively.

Location-based billing System: GSM operators have different billing methods for different users/subscribers. In Nigeria, GSM operators like Etisal at Nigeria Ltd and MTN Nigeria Ltd have a billing system called “home zone”. When a mobile user registers for the home zone billing platform, the GSM operator charges a lower fee for calls made by a mobile user within the geographical area/zone that the mobile user registered as his home zone while charging a different/higher fee for calls made by the same mobile user in any other geographical area/zone other than the area registered as the home zone. This location-based billing system is made possible because the moment a mobile user registers for the home zone billing system, the mobile user number is setup with a system that continuously queries and monitors the location of the user and the answer to this continuous query determines the billing rate that will be used for the user at every point in time and location.

Location-dependent Advertisement: In the field of mobile commerce, the concept of location-dependent advertising, or proximity-triggered advertisements, arises as an effective way of attracting nearby users to stores [1]. Thus, for example, people outside a supermarket could receive e-discounts that encourage them to shop inside [3]. Supermarket owners sometimes send interesting offers to GSM phone users that are near to the supermarket coverage areas. Also, politicians employ this technology to send campaign messages to prospective voters that are within a geographical location; for instance, in a city, state, or country.

This type of advertisement is made possible because there is a system that continuously monitors and queries the presence of a mobile user within a particular or defined coverage area. Navigation and Information Services: A mobile user or a person carrying a portable digital assistance (PDA) with GPS capabilities is able to obtain information about is current location. As the mobile user arrives in a new location, he/she gets updates about his new location. Digital travel assistants, location-dependent yellow pages, Google earth, etc., are powered by systems with continuous query components [2].

Alarm Systems: Various alarm systems are available because they employ continuous query component of the application to determine when a set condition has been met and thus, triggers the required alarm which could be either audible or visual in nature. Alarms could be timer-based or change based. A simple timer-based alarm system is found in a wall clock. A clock user can set a "wake up" audible alarm to trigger at 0400 hours every day. In the process and automation industries, change-based alarm systems are used to indicate when there are process upsets and emergency situations in the process plant. For instance, an aircraft cockpit is equipped with alarm systems that notify the pilot of possible faults or dangers. In a gas processing plant, excess pressure, high temperature, or excess flow rate, etc., can lead to severe consequences. These parameters are continuously monitored against set points and once there is a change or increase in the parameter that indicates offset, an audible or visual alarm is activated to notify the control room operator that a particular parameter set point is exceeded.

IV. CONCLUSION

We have introduced continuous query, its major classifications, and some application areas of continuous queries. Location-based service (LBS) applications, online transaction processing (OLTP) applications, alarm systems, etc. are strong evidences of the role of continuous query in the emerging data stream management system. The benefits of these applications in our day-to-day activities justify the essence of this research.

V. ACKNOWLEDGEMENT

I am deeply indebted to my supervisor, Prof. E. O. Nwachukwu of the Department of Computer Science, University of Port Harcourt, for his fatherly advice, visionary suggestions and encouragements while this research effort lasted. Also, I am grateful to the Head of Department of Computer Science, University of Port Harcourt, Dr. L.N Onyejegbu, Dr. Charles Ugwu, and the entire staff of Computer Science Department, University of Port Harcourt, for providing all the necessary and enabling academic supports for the success of this Paper. Special thanks go to Dr. Bartholomew Eke of the department of Computer Science, Uniport, for his time and technical assistance.

REFERENCE

1. Aalto, L., Gothlin, N., Korhonen, J., and Ojala, T.(2004). Bluetooth and WAP push based location-aware mobile advertising system. In Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys). ACM Press, New York, NY, 49–58.
2. Chon, H. D., Agrawal, D., and Abbadi, A. E. (2001). Using space-time grid for efficient management of moving objects. In Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE). ACM Press, New York, NY, 59–65.
3. Ilarri, S., Mena, E., and Illarramendi, A. (2010). Location-dependent query processing: Where we are and where we are heading. ACM Computing Surveys, 42, 3, Article 12, ACM Press, New York.
4. Jianjun Chen, David J. DeWitt, Feng Tian, and Yuan Wang (2000).
5. NiagaraCQ: A Scalable Continuous Query System for Internet Databases, Proceedings of the ACM-SIGMD International Conference on Management of Data, May 16-18, 379-390, Dallas.
6. Liu L., Pu C, and Tang W. (1999). Continual Queries for Internet Scale
7. Event-Driven Information Delivery. TKDE 11(4): 610-628, IEEE Transactions on Knowledge and Data Engineering (TKDE).
8. Terry D., D. Goldberg, D. Nichols, and B. Oki. (1992) Continuous Queries
9. Over Append-Only Databases. Proceedings of the ACM-SIGMOD International Conference on Management of Data, 321-330, New York.

