ABSTRACT:

Continuous queries are used to monitor changes to time varying data and to provide results useful for online decision making. Typically a user desires to obtain the value of some aggregation function over distributed data items, for example, to know value of portfolio for a client; or the AVG of temperatures sensed by a set of sensors. In these queries a client specifies a coherency requirement as part of the query. We present a low-cost, scalable technique to answer continuous aggregation queries using a network of aggregators of dynamic data items. In such a network of data aggregators, each data aggregator serves a set of data items at specific coherencies. Just as various fragments of a dynamic web-page are served by one or more nodes of a content distribution network, our technique involves decomposing a client query into sub-queries and executing sub-queries on judiciously chosen data aggregators with their individual sub-query incoherency bounds. We provide a technique for getting the optimal set of sub-queries with their incoherency bounds which satisfies client query's coherency requirement with least number of refresh messages sent from aggregators to the client. For estimating the number of refresh messages, we build a query cost model which can be used to estimate the number of messages required to satisfy the client specified incoherency bound. Performance results using real-world traces show that our cost based query planning leads to queries being executed using less than one third the number of messages required by existing schemes.

The Greedy algorithm for query plan selection

Result ← Ø

While A ≠ Ø

Choose a sub-query a ϵ *A with criteria* ψ

Result ←*Result* υ *a*

A ← *A-{a}*

For each data element e ϵ *a*

For each b ϵ *A*

b ← *b-{e}*

If b = Ø

A ← *A-{b}*

Else

Calculate sumdiff for modified b

Return Result

Existing System:

Many data intensive applications delivered over the Web suffer from performance and scalability issues. Content distribution networks (CDNs) solved the problem for static content using caches at the edge nodes of the networks. CDNs continue to evolve to serve more and more dynamic applications. The static fragments are served from the local caches whereas dynamic fragments are created either by using the cached data or by fetching the data items from the origin data sources. One important question for satisfying client requests through a network of nodes is how to select the best node(s) to satisfy the request. For static pages content requested, proximity to the client and load on the nodes are the parameters generally used to select the appropriate node.

Disadvantage:

1. For data item which needs to be refreshed at an incoherency.
2. The exact data value at the corresponding data source need not be reported as long as the query result satisfies user specified accuracy requirements.

Proposed System:

Continuous queries are used to monitor changes to time varying data and to provide results useful for online decision making. This paper we present a low-cost, scalable technique to answer continuous aggregation queries using a content distribution network of dynamic data items.

Advantage:

1. It saves the time and the user spending low cost.
2. A continuous query cost model which can be used to estimate the number of messages required to satisfy the client specified incoherency bound.
3. We present to implementations of Continuous Aggregation in optimized query.

MODULES

1. Security Module
2. Flow chat Module
3. Update Module
4. Client/Server Module

Security Module:

This module is used to help the user to provide the security of access. Because once the user to logout or leave our account automatically user password is changed and server to send the password in our mail ID. Whenever the user to logout the account automatically the security key is changed based on the random function.

Flow Chat Module:

This module is used to help the user to view the BSE and NSE value in bar flow chat based on the date. This chat to display the aggregated value based on the companies sharing values continuously. The companies value is changed automatically chat value is changed.

Update Module:

This module is used to help the user to view the BSE and NSE value to update in every minute. So the user waiting time is reduced and sees the updated value in every minute. The server to set the time when our form is updated.

Client/Server Module:

This module is used to help the client and server interaction to the database. This module is used to dynamically create the table based on the server entering value. These values are assigning to the chat x and y position and display the client. These values are changed in dynamically based on the server entering values.

SYSTEM SPECIFICATION

Hardware Requirements:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 14’ Colour Monitor.
- Mouse : Optical Mouse.
- Ram : 512 Mb.
- Keyboard : 101 Keyboard.

Software Requirements:

- Operating system : Windows XP.
- Coding Language : ASP.Net with C#
- Data Base : SQL Server 2005.