

Adaptive query planning procedures to support Continuous Aggregation Queries

Panidapu LakshmiSivaSaila¹, J.Ranga Rajesh², T.Hema Latha³

¹CSE Dept, Devineni Venkata Ramana & Dr Hima Sekhar MIC College of Technology , Vijayawada

²Assistant professor, Department of CSE Dept, Devineni Venkata Ramana & Dr Hima Sekhar MIC College of Technology, Vijayawada

³Assistant professor, Department of Computer Science and Engineering, KL University, Guntur Dist,AP

ABSTRACT: Continuous queries are persistent queries that can transform a passive web into an active environment by providing time varying dynamic query results useful for online decision making. To handle countless with different hobbies, a constant inquiry framework must be equipped for supporting server push style of Internet-based correspondence. For versatile treatment of push based information dispersal, former methodologies utilized a system of information aggregators. Their usage obliged Greedy Heuristics Algorithm alongside preconfigured incoherency limits to oversee both numerous aggregators and various customers for supporting server push based correspondences. Existing heuristic-based methodologies can just investigate a constrained arrangement space and subsequently may prompt imperfect arrangements. So we propose to utilize a versatile and expense based methodology. In a system of information aggregators, each one committed and sensibly picked aggregator serves a set of information things at particular coherencies. It includes disintegrating a customer question into sub-questions and executing sub-inquiries utilizing aggregators with their individual sub-inquiry incoherency limits. Our expense model considers both the preparing expense and the correspondence cost not at all like former methodologies. Versatile and expense based methodology has better execution regarding both preparing and correspondence cost than plain Greedy

Heuristics methodology and a useful usage approves the proposed case.

Index Terms: Greedy heuristics algorithm, aggregation, queries, process message, Greedy Operator Ordering (GOO).

2. INTRODUCTION

Applications, for example, barbers, individual portfolio valuations for budgetary choices, sensors-based checking, course arranging focused around activity data, and so on., make broad utilization of element information. For such applications, information from one or more free information sources may be amassed to figure out whether some activity is justified. Given the expanding number of such applications that make utilization of exceptionally dynamic information, there is critical enthusiasm toward frameworks that can effectively convey the applicable upgrades naturally. As a case, consider a client who needs to track an arrangement of stocks in distinctive (business) accounts. Stock information values from conceivably distinctive sources are obliged to be collected to fulfill client's necessity. In these ceaseless question applications, clients are liable to endure some mistake in the results. That is, the careful information values at the comparing information sources require not be accounted the length of the question results fulfill client defined precision necessities.

Information Incoherency: Verifiably the refinement in estimation of the data thing at the data

source and the quality known to a client of the data. Let $v_i(t)$ show the estimation of the i th data thing at the data source at time t . The quality the data thing known to the client be $u_i(t)$. Yet the dynamic confusion at the client is given by $|v_i(t) - u_i(t)|$. At the point when a data incoherency surpasses C the data strengthen message is sent to the client for the data thing. i.e., $|v_i(t) - u_i(t)| > C$. Arrangement of Data Aggregators (DA): Using push- or draw based frameworks the data empower from data sources to clients is conceivable. In the power based part data sources send messages to the client exactly when the client makes sales whereas in the push-based instrument data sources send upgrade messages to clients all alone. For the data trade between the source and the client we suggest push- based instrument. For versatile treatment of push based data dispersal, arrangement of data aggregators are proposed as that dissipating tree from sensor centers to root starting now exists; and they moreover present failure channels on partly adds up to. Data animates happen from data sources to the clients through one or more data aggregators. We acknowledge that each data aggregator keeps up its organized incoherency limits for distinctive data things. In data spread a different leveled arrangement of data aggregators is used such that each data aggregator serves the data thing at some guaranteed incoherency bound. The data dispersing capacity point of view, each data aggregator is depicted by an arranged of (d_i, c_i) sets. Where d_i =data thing c_i =incoherency bound.

3 LITERATURE REVIEWS

3.1 Disseminating Streaming Data in a Dynamic Environment [B.Chin Ooi,The VLDB Journal, vol. 17,2008].

In this paper, we reconsidered the issue of planning a versatile spread framework. We proposed an expense based methodology to develop spread trees to minimize the normal loss of constancy of the framework. Two static calculations: Greedy and SA, have additionally been proposed for generally static situations and for building beginning trees under dynamic situations. The Greedy calculation is valuable for element situations because of its speedier rate to manufacture a generally decent introductory tree, while SA is better for static situations due than its heartiness. Moreover, the multi-tree methodology is indicated to be more powerful to the quantity of articles, the level of information enthusiasm and additionally framework workload.

3.2 Efficient Constraint Monitoring Using Adaptive Thresholds [R.Rostogi and P.Shukla IEEE 24th International Conference, April 2008].

In this paper, we have considered the execution of a basic appropriated stipulation, total of variables. It would be intriguing to sum up the perception that non-zero slack techniques can bring about better execution for general capacities (like join sizes, quintiles and so forth.) utilizing the structure .A novel following issue called aggregate triggers and it would be fascinating to perceive how our systems perform when connected to their issue. In average systems, hubs can be sorted out in a various leveled structure that can be misused to further decrease correspondence needed in actualizing disseminated requirements. Mulling over non-zero slack calculations for such organized systems exhibits an intriguing zone of future research as well.

3.3 Executing Incoherency Bounded Continuous Queries at Web Data Aggregators. [R.Guptha , International world wide web conference, May 10 ,2005].

In this paper we considered the issue of noting questions for online choice making at web information aggregators. We demonstrated that such questions can be spoken to as consistent inquiries with limited incoherency and loyalty prerequisites. Through a probabilistic information model and input based component we demonstrated that our methodology can be productively used to convey inquiry results with customer's coherency and devotion necessities. A paramount part of our methodology is the utilization of existing web framework for noting these inquiries which prompts negligible structural prerequisites and more adaptability.

EXISTING SYSTEM

Continuous queries are persistent queries that allow users to receive new results when they become available. While continuous query systems can transform a passive web into an active environment, they need to be able to support millions of queries due to the scale of the Internet. continuous queries allow users to obtain new results from a database without having to issue the same query repeatedly. For example, users might want to issue continuous queries of the form: *"Notify me whenever the price of Dell or Micron stock drops by more than 5%and the price of Intel stock remains unchanged over next three month."* For supporting continuous queries for the users, the service provider is maintained and managed by a single resource builder. 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 resident data storages. So a better system is required to support continuous queries for the users, at a time where the system is

maintained and managed by a multiple resource builders using network aggregators.

Problem Statement

Continuous queries are persevering inquiries that permit clients to get new comes about when they get to be accessible. While consistent inquiry frameworks can change a latent web into a dynamic environment, they have to have the capacity to help a great many questions because of the scale of the Internet. Ceaseless questions permit clients to get new comes about because of a database without needing to issue the same inquiry more than once. For instance, clients may need to issue persistent inquiries of the form: "notify me at whatever point the cost of Dell or Micron stock drops by more than 5%and the cost of Intel stock stay unaltered over next three month". For supporting ceaseless questions for the clients, the administration supplier is kept up and oversaw by a solitary asset manufacturer. To handle an extensive number of clients with assorted hobbies, a nonstop question framework must be fit for supporting countless communicated as unpredictable inquiries against inhabitant information stockpiles. So a superior framework is obliged to backing persistent questions for the clients, at once where the framework is kept up and oversaw by a various asset manufacturers utilizing system aggregators.

4 PROPOSED SYSTEM

Here we used aggregation weights to update the results to clients. By aggregation weights we can maintain the processing time.

We take the response time and request time to maintain query processing time.

Response time= time between initiation and completion of the query.

Request time = CPU time + I/O time (in nano sec).

Query processing= reqtime + restime.

In proposed we use aggregation function by using HTTP protocol "HTTP-BIO-8084-EXEC-29", for every exam has its own function. By using this function the searching time in the database is reduced.

Whenever the client send the request aggregator responded immediately by accessing the HTTP protocol. Main reason to use this function is , the client gives the query in different formats in any format this protocol takes and access the query.

By using adaptive approach we can access the client query in any situation either economically or technically.

Push, or server push, depicts a style of Internet-based correspondence where the appeal for a given exchange is started by the distributor or focal server. It is appeared differently in relation to force, where the appeal for the transmission of data is started by the recipient or customer. Utilizes server push based systems for starting interchanges. Push administrations are regularly focused around data inclination communicated ahead of time. This is known as a distribute/subscribe model. A customer may "subscribe" to different data "channels". At whatever point new substance is accessible on one of those channels, the server would push that data out to the client. For versatile treatment of push based information scattering, we utilize a system of information aggregators. Information invigorates happen from information sources to the customers through one or more information aggregators. Heuristic alludes to experience-based procedures for critical thinking, learning, and disclosure. Utilizes Greedy Heuristics Algorithm alongside preconfigured

incoherency limits to oversee both different aggregators and numerous customers accordingly conveying a finer execution.

Earlier Approaches use Greedy Heuristics Algorithm alongside preconfigured incoherency limits to oversee both numerous aggregators and various customers for supporting server push based correspondences. Question advancement methods created utilizing Greedy Heuristics Algorithm relies on upon transforming cost just. Existing heuristic-based methodologies can just investigate a restricted arrangement space and consequently may prompt problematic arrangements. So we propose to utilize a versatile and expense based methodology. Our expense model considers both the preparing expense and the correspondence cost. Versatile and expense based methodology execution includes

- Adaptation Attempt (to check for feasibility)
- Greedy Heuristics
- Simulated Annealing
- Process Message

Versatile and expense based methodology has better execution as far as both preparing and correspondence cost than plain Greedy Heuristics approach.

QUERY AGGRIGATION AND THEIR EXICUTION

To execute in incoherency constrained steady question course of action is required. We show a strategy for executing multi-data aggregate inquiries. The subject of our arrangement is to lessening the amount of resuscitate messages from data aggregator to client. For the better understanding take after the circumstances.

Scenario 1: a. assume the query $Q=60x_1+190x_2+150x_3$, Where X_1, X_2, X_3 are data things for stock with incoherency bound of \$75.

For the considered circumstance the client can get the results as a. among information things Client can get data things autonomously on request incoherency bound is divided.

b. the request a single data aggregator can pass on to all data things to answer.

c. a single request may be secluded into number of sub- inquiries and emerge data aggregator gives their qualities.

Scenario 2: for example the sensor framework and an AVG request over the target set of sensors (d_1, d_2, d_3) imbued at the inquiry center. Amassing is used inside the frameworks for the beneficial essentialness spread of sums. Interface the target sensors and inquiry centers for creating the aggregation tree, each center can pick the path to the request center point concentrated around the slant segment. We have to pick the aggregate path in the framework such that we can execute minimum number of messages.

The essential arrangement of the circumstances is

a. generating the sub-questions from the request

b. the Incoherence Bound is consigned

c. the sub-inquiries are executed at the picked data aggregates.

d. reducing in the restoring messages.

The total of the execution cost of sub-inquiries is just number of strengthens.

```

result ← ∅
while  $M_q \neq \emptyset$ 
    choose a sub-query  $m_i \in M_q$  with criterion  $\psi$ :
    result ← result  $\cup$   $m_i$ ;  $M_q \leftarrow M_q - \{m_i\}$ 
    for each data item  $d \in m_i$ 
        for each  $m_j \in M_q$ 
             $m_j \leftarrow m_j - \{d\}$ ;
            if  $m_j = \emptyset$   $M_q \leftarrow M_q - \{m_j\}$ ;
            else calculate sumdiff for modified  $m_j$ ;
return result
    
```

Fig 1: Greedy heuristic data dissemination process model.

Subsequently the result is acquired to utilize a versatile and expense based methodology.

Input: a set of relations to be joined and weight function

Output: a join order

```

S = 0
while ( $|R| > 0$ )
{
     $m = \arg \min R_i \in R$ 
     $w(R) \ R = R \setminus \{m\}$ 
     $S = S \cup \langle m \rangle$ 
}
    
```

Greedy Heuristics approach

- The previous greedy algorithms only construct left-deep trees
- Greedy Operator Ordering (GOO) constructs bushy trees

Idea: Consolidate joins trees such that the middle of the road result is insignificant, where

relations have to be joined some place yet joins can likewise happen between entire join trees...

Costs: The costs for a totally ordered precedence graph G can be computed as follows:

$$C_H(G) = \sum_{i=2}^n [n_{1,2,\dots,i-1} h_i(n_i)]$$

$$= \sum_{i=2}^n [(\prod_{j=1}^i s_j n_j) h_i(n_i)]$$

EXPERIMENTAL RESULTS

Existing heuristic-based approach can simply examine a limited result space and thusly may incite sub- perfect results. Flexible and cost based technique has better execution the extent that both planning and correspondence cost than plain Greedy Heuristics approach. We display the reenactment comes to fruition on request getting prepared for growing the framework execution in genuine word component.

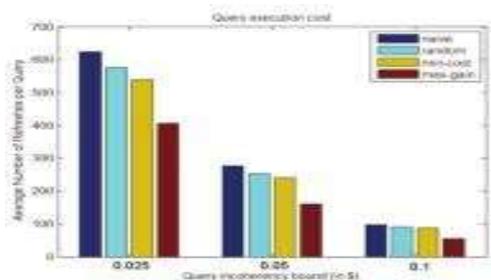


FIGURE 2: COMPARISON OF GREEDY ALGORITHM WITH HEURISTIC RESULTS.

For weighing the likelihood in framework, we have to use gathering attempts inside the framework concerning inquiry taking care of and correspondence process organization in the nature's space.

CONCLUSION:

Ceaseless questions are industrious request that allow customers to get new happens when they become

available. The organization supplier is kept up and administered by a single resource producer for supporting steady inquiries for the customers. Remembering the deciding objective to handle incalculable with different preoccupations, a steady request schema must be prepared for supporting a far reaching number of triggers conveyed as unpredictable request against occupant data stockpiles. This will be known as a disseminate/subscribe model. At whatever point new substance is open on one of those channels, the server would push that information out to the customer. So we propose to use an adaptable and cost based procedure. Our cost model considers both the changing cost and the correspondence cost. Adaptable and cost based strategy has better execution in regards to both taking care of and correspondence cost than plain Greedy Heuristics approach.

REFERENCES:

- [1] "Maintainin Coherency of Dynamic Data in Cooperating Repositories," S.Shah, K. Ramamritham, and P. Shenoy, Proc. 28th Int'l Conf. Very Large Data Bases (VLDB), 2002.
- [2] "Query Planning For Continuous Aggregation Queries Over A NetworkOf data Aggregators" Rajeev Gupta, Kirthi Ramaritham, vol 24, No. 6. June 2012.
- [3] "Disseminating Streaming Data in a Dynamic Environment: An Adaptive and Cost Based Approach," Y.Zhou, B. Chin Ooi, and K.-L. Tan, The Int'l J. Very Large Data Bases, vol. 17, pp. 1465-1483, 2008.
- [4] "Construction of aTemporal Coherency Preserving Dynamic Data Dissemination Network," S. Agrawal, K. Ramamritham, and S. Shah,

Proc. IEEE 25th Int'l Real-Time Systems Symp. (RTSS), 2004.

[5] "Executing Incoherency Bounded Continuous Queries at Web Data Aggregators", R. Gupta, A. Puri, and K. Ramamritham, Proc. 14th Int'l Conf. World Wide Web (WWW), 2005.

[6] "Adaptive Filter for Continuous Queries over Distributed Data Streams", C. Olston, J. Jiang, and J. Widom, Proc. ACM SIGMOD Int'l Conf. Management of Data, 2003.

[7] "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks", S. Madden, M.J. Franklin, J. Hellerstein, and W. Hong, Proc. Fifth Symp. Operating Systems Design and Implementation, 2002.

[8] "Computers and Intractability-A Guide to the Theory of NP-Completeness", M.R. Garey and D.S. Johnson, V Klee, Ed. Freeman, New York, 1979.