ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A Review on Data Warehouse Management

Umair Rasheed^{*1}, M.Umer Sarwar^{2}, Ramzan Talib^{3}
^{*1,2,3} College of Computer Science & Information Studies, Government College University, Faisalabad, Pakistan
umair514@gmail.com

Abstract

Data warehouse management is a crucial part of industry and business that have been adopted and put into practice with increased complexity of managing pools of data. The architectural layout of data warehousing is composed of a centralized warehouse from which views are generated for needy users. The convention in this case is that the warehouse receives an information injection from individual distributed databases which may as well be not related to the warehouse but hold the raw data needed to justify the existence of warehouse. However, data warehouse management is a relatively complex procedure that needs data updates from distributed databases. The complexity of this procedure relates to anomaly-prone informational transaction between the data source end and the materialized view end. This description supports the critical requirements of data warehouses which are accuracy and timeliness. This review dwells on the assessment of data warehouse literature to verify the dominant strategy employed in data warehouse management and the trade-offs involved. For this purpose various data warehouse management related articles were selected and carefully reviewed. The findings revealed a favor towards Immediate Incremental Management (IIM) of warehouse against Deferred Incremental Management (DIM). It is observed that system availability which constitutes accurate views and real-time updating was the base determining factor. Deferred Incremental Management has been the conventional way but with an increase in data volume that requires a gigabyte transaction rate that defines contemporary business and industry this updating mechanism is quite unsuitable and strips data warehouse management its contextual meaning.

Keywords: Immediate Incremental Management (IIM), Deferred Incremental Management (DIM), Algorithm, Informational transaction.

Introduction

Data warehousing has been prevalent in contemporary business and industry and be described as the de facto informatics of the time. It has been a prerequisite for conducting information-oriented business and various industrial functions. Intelligence informatics has been applied in these two environments by use of data warehousing to create a discernible culmination of events and generate logically understandable summary information critical for decision-making [1]. The term “data warehouse” traces root back to the 1980s and has seen adaptation and structural evolution since then [2]. The terms carried a general meaning of the nature of basic system behind the process and the purpose of decision making it supported. Recent advancement however has given data warehousing a comprehensive meaning solely related to time and the revolutionary changes involving information-flow protocol and transaction guidelines.

Data warehouse is particularly a repository of integrated information that has been gathered from distributed sources often databases [2, 3, 4, 5]. The information is bundled up by dedicated hardware and software systems that subject the information (data) to an Extraction-Transfer-Load schema by implementing algorithms and definite schemas optimized for the process [6]. This enables logical representation of juggled machine-oriented data within the warehouse system to relevant viewers. Since it is vital for data warehouse management to consider timing and accuracy of materialized views and it is imperative to assess the optimum mechanism of upholding user-expected time while accessing warehoused information in the user-expected accuracy. This review assumes Immediate Incremental Management as the optimal strategy in achieving accuracy and timing in fast and reliable data warehouse management. It also assumes concurrent exchanges between the source and the viewers is optimally transacted on the fly rather than apportion a downtime against warehouse access which locks out non-admin access.

Materials and Methods

This paper is based on the factual extracted information from randomly selected journal articles in the field of Information Systems. The articles were initially subjected to abstract analysis to ascertain their relevance in information gathering. Twenty-two articles were positively identified fit for the study and thoroughly read wholly to further ascertain their credibility for the review. All articles were subjected to a text search to verify the inclusion of keywords and phrases that characterize the review and stipulated hypotheses. Out of the 22 three were particularly shallower than the rest 19 due to the presence of an abstract and an introduction without detailed figurative data. They were however included for inclusion of critically needed facts in the introduction and their abstract tendency to positively engage the issue of data warehouse management. The review however took particular interest in ten of the articles due to direct relationship of researched information and the review's hypotheses. Calculations included in these journal articles were carefully rerun to guarantee information accuracy and consistency throughout the paper. Figurative data was verified to be authentic and the review delved into the articles to collect needed information to challenge the hypotheses.

Result

Modernized data access is run from systems that implement fast and parallel information transactions to give user a real-time experience with information [7]. Systems have been evolved to embrace a parallel communication system between data source end and material viewer end. Immediate Incremental Management (IMM) occasionally referred to Immediate Incremental View Maintenance (IIVM) is the current most relevant mechanism to implement in data warehouse management besides from scratch re-computing. IMM/IIVM is preferred for its resilience and better handling of anomalies related to online-analysis process (OLAP) queries. A majority of the reviewed literature which further suggests different algorithms to realize the full potential of the technique has affirmed this result.

Discussion

Architecture

Data warehouse management involves a transactional technique to communicate between the different infrastructural modules involved [8]. The transactions employ a handshake protocol between the specified modules to realize a proper data placement with the right confirmation notification. A typified model suggested by Warehousing Information Project at Stanford (WHIPS) outlays the modular architecture that characterizes conventional data warehousing [9]. The model composes a source module (the various distributed databases from which information originates), monitor wrapper module (tasked with monitoring changes at the source level and relaying notice to the integrator by use of trigger-based technique by use of system query language), integrator module (receives source-related updates and forwards the to view managers), view manager module (every viewer is assigned a manager and utilizes a strobe algorithm to guarantee consistency), a warehouse wrapper (translate information to a warehouse discernible signal), warehouse module and a warehouse application module (the interfacing infrastructure accessible to information seekers). At users disposal are data sources, system logs and warehouses involved [9]. According to [10] there are three types of implementation architectures for a data warehousing, which are single-layer, two-layer and three-layer having their own strengths and weaknesses.

Data flow

The consistent and most defining factor of data warehouse management is the organization of data. A simplified explanation towards the concept of data warehouse management is proposed in [10] which imply complete and consistent store of data collected from a variety of sources and made available to needing users. The major consideration in this fact is the neediness of accessing data. The infrastructural working in the warehouse ensures this needed data is captured, processed and relayed to the user in a logical, understandable way free from errors and at the right time. Data movement in the warehouse is an important and indispensable aspect in this review and understanding the flow of data from the source through the infrastructure is a vital step to discerning the results and what they imply.

Data obtainable from distributed databases (often involving a particular industry) is transferable to the user through three main components the source (which contains needed data in raw form), the warehouse infrastructure (which processes the data) and the access application (the user-warehouse interface). This generalized method is referred to as Extraction, Transfer and Load (ETL) according to process [6]. Extraction is reading the data from the source in its raw form and feeding it. Reading is executed at the monitor wrapper interface obtains data deemed updates (augmentation), deletion or subtraction and the reading command picks the necessary query trigger to make the opted changes (SQL). Transfer involves the infusion of the data from the source into the warehouse infrastructure where organizational commands are initiated to adjust materialized view accordingly before availability to the needing user through the Load procedure.

Algorithmic implementation

Concurrency and parallel exchange in schematic flow of a data warehouse system has been an issue of concern that has played a big setback in maximizing the potential of many data warehouses observed in a programming model exemplified by [11]. The abstracted rapid exchange of information between two database sources while the warehousing system is still making necessary changes to previous summary logs is prone to inconsistencies and errors in final tables and unfortunately manifest to users. This calls for yet another wasted minute to make necessary changes and maybe by then the inconsistencies and errors would have already taken effect on users thought the dynamic materialized views. This sort of system is highly unacceptable and is non-adaptive to present day systems that are modeled to operate under an up-to-date sentiment in influencing just-in-time decisions [5].

To counter these system flaws certain software structures are needed to implement sustainable and dependable system behavior. Conventional procedures that help shield these transactional inconsistencies and anomalies have been put across and include SWEEP for source data verification, View Adaptation (VA) to monitoring view changes and Evolvable View Environment (EVE) for schema change monitoring at the information source. These integrative procedures invoke relevant commands within the whole system to take necessary action when their reading changes. It is done on a change-trigger basis to outdo timing and accuracy. These three micro-algorithms are infusible into a comprehensive algorithmic structure referred to as Schema change and Data update Concurrency Control system dubbed SDCC. This all-inclusive algorithm is implemented with accuracy and timing in mind factors which are stressed by the Immediate Incremental Management of data warehouses. SDCC employs Local Compensation (LC) that verifies every information node before being translated into the next to counter the problem of error spillage to other non-affected system modules [11]. This improvised LC quickly corrects faulty queries before they enter the mainstream of data transfer.

The LC operates under SDCC command and is invoked and “killed” by it. In the event of faulty query detection normally resulting from concurrencies gone wrong SDCC locates the query gathers information about the faulty-query result and the maintenance-concurrent updates. LC algorithm solves the problem through a four-leveled correction. First procedure is creation of a temporary relational table. Second procedure is the creation of one local-compensation query. Third procedure is ascertaining the delta (change) in query. Fourth procedure is obtaining the value transition by subtracting the query delta from the created local-compensation query derived in the second procedure [11].

Analysis

Data execution in the warehousing environment is prone to errors in the contemporary business and industry sector which are prevalent customers that use this utility. Technological advancement has raised the bar of data transmission to terabytes per second. Online platforms as well have been submerged into using data warehousing management to facilitate easy access of information [14]. Since the globalization and worldwide access to information has been the norm of the time and data warehouse downtime that has been characteristic of the dawn-of-connectivity warehouses has ceased to be efficient and have been deemed inefficient and rather pose as impediments than utilities [6]. Downtime is the deactivation of user-access of the data warehouse as a management effort to make necessary adjustments like cache refreshments and re-computing tables. Downtime however as much as it is planned and executed in the best interest of the utility holds negative consequences of locking users out of the system without much notice. Personalized contact with information therefore is not guaranteed on a full time basis. Even though warehouse managers claim the process to execute this duty when user access is virtually offline but some disagree with this fact [15].

Global accessible data warehouses that are dominant in internet applications and form characterized websites need cannot whatsoever use Deferred Incremental Management data warehouses due to constant global access. Data warehouses useable in this instance therefore employ an Immediate Incremental Management system that properly suits this environment. Globally accessible data warehouses cannot be justified to have a possible downtime since a part of the world is accessing the data. These interchangeable patterns keep revolving and therefore need an increment on the fly [16].

Suggested and verified in this course is an Immediate Incremental Management system that makes progressive updates with dedicated algorithms. IIM/IIVM is the present day implemented system since most data warehouses need one way or another a global connection. Information collaboration has also pushed this need for IIM/IIVM to carry out long stretch multinational projects. With globalization for instance and the need to abate climate deterioration worldwide information collection in a centralized data warehouse is much needed than before to enhance the likelihood of succeeded in such a course [17].

Conclusion

Data warehouse management is a contemporary issue that has defined intelligence and advancement in the field of business industry. Technological advancement has seen more researchers, analysts, business and industry personnel need information to demarcate trends to plan the future and make localized decisions. However not only these but the common persons have also gained interest in data. This has generated much traffic that has led to infrastructural hitches. On the other hand the need has arisen to increase the availability of data warehouses to the benefit of those that need its services. IIM has been selected as the best possible way of algorithmic implementation with little compromise to either the management team or the users. Batch processing implemented in Deferred Incremental Management was a likely option but quite disadvantageous and irrelevant to contemporary times. IIM on the other hand was observed to pose a better chance of making fast corrective measures. SDCC local sub-querying system for example demonstrated this principle of accurate correction which can run simultaneously in an (n) number of Information Sources. The “n” in this case representing the possible number of distributed databases that communicate information to the centralized data warehouse [20].

Different algorithms have been suggested however but with no in-depth analysis. This review paper took to the bottom of Schema change and Data update Concurrency Control system (SDCC) to explain a portion of the comprehensive operation of a sub-level algorithm. Various algorithmic expressions however are implementable within the same functionality of LC. DIM employs a different strategy different from IIM. Operations regarding the system for instance critical augmentations, subtractions, deletions and subsequent translation to summary logs are not that possible on a real time basis which is quite a setback to a time conscious business and industry environment that is dominated by fast updating and just-in-time decisions. Internet enabled data warehousing environments which dominate the contemporary society either will definitely not need DIM and the reason behind this is the effect of downtime [21].

DIM cannot be tested against IIM when time and efficiency is a factor. IIM related algorithms as well form the optimum procedure in countering errors generated during concurrent information transaction between the user side accessing the data warehouse material views and the information sources/databases interfacing with the system to feed data. Accuracy and timeliness are the discriminating parameters that characterize data warehouse management. However, downtime-prone warehousing management system may be used in non-time conscious data management environment that does not need constant access [22].

References

[1]. *Elamy, A.H.; Alhajj, R.S.; Far, B.H., "Building data warehouses with incremental maintenance for decision support," Electrical and Computer Engineering, 2005. Canadian Conference on, pp.1809-1814, 2005.*

[2]. *Saeki, S.; Bhalla, S.; Hasegawa, M., "Parallel generation of base relation snapshots for materialized view maintenance in data warehouse environment," Parallel Processing Workshops, 2002. Proceedings. International Conference on, pp.383-390, 2002.*

[3]. *Zhang, X.; Rundensteiner, E.A., "Data warehouse maintenance under concurrent schema and data updates," Data Engineering, 1999. Proceedings., 15th International Conference on, pp.253, 1999.*

[4]. *Lee, J.W.T.; Xiang Ye, "Materialized view design and maintenance in a financial data warehouse system," Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on, vol.5, pp.930-935, 1999.*

[5]. *Ismail, R.M., "Maintenance of materialized views over peer-to-peer data warehouse architecture," Computer Engineering & Systems (ICCES), 2011 International Conference on, pp.312-318, 2011.*

[6]. *Wei Xu; Maoqing Li; Shunxiang Wu; Shunzhi Zhu; Zhoujing Wang; Kehua Miao; Ying Wang, "Incremental data feed maintenance of a data warehouse system derived from multiple autonomous data sources," Control and Automation, 2005.*

ICCA '05. International Conference on, vol.2, pp.1108-1113, 2005.

[7]. Yeu Shiang Huang; Do Duy; Chih Chiang Fang, "Efficient maintenance of basic statistical functions in data warehouses," Decision Support Systems, vol.57, pp.94-104, 2014.

[8]. Chao C, "Incremental maintenance of object-oriented data warehouses," Information Sciences, vol.160, no.1-4, pp.91-110, 2004.

[9]. Wiener, J.L.; Gupta, H.; Labio, W.J.; Yue Zhuge; Garcia-Molina, H.; Widom, J., "The WHIPS prototype for data warehouse creation and maintenance," Data Engineering, 1997. Proceedings. 13th International Conference on, pp.589, 1997.

[10]. Chan, A.; Crane, G.; MacGregor, I.; Meyer, S., "Data warehouse on the Web for accelerator fabrication and maintenance," Particle Accelerator Conference, 1997. Proceedings of the 1997, vol.2, pp.2413-2415, 1997.

[11]. Zhang, X.; Rundensteiner, E.A., "The SDCC framework for integrating existing algorithms for diverse data warehouse maintenance tasks," Database Engineering and Applications, 1999. IDEAS '99. International Symposium Proceedings, pp.206-214, 1999.

[12]. de Amo, S.; Alves, M.H.F., "Efficient maintenance of temporal data warehouses," Database Engineering and Applications Symposium, 2000 International , pp.188-196, 2000.

[13]. Buren, G.; Ruckert, C., "Architectural Maintenance Using a Data Warehouse System for Availability Analysis," Software Maintenance and Reengineering, 2009. CSMR '09. 13th European Conference on, pp.307-308, 2009.

[14]. Kulkarni, S.; Mohania, M., "Concurrent maintenance of views using multiple versions," Database Engineering and Applications, 1999. IDEAS '99. International Symposium Proceedings, pp.254-258, 1999.

[15]. Ram, P.; Do, L., "Extracting delta for incremental data warehouse maintenance," Data Engineering, 2000. Proceedings. 16th International Conference on, pp.220-229, 2000

[16]. Reddy, S.S.S.; Lavanya, A.; Khanna, V.; Reddy, L. S S, "Research Issues on Data Warehouse Maintenance," Advanced Computer Control, 2009. ICACC '09. International Conference on, pp.623-627, 2009.

[17]. Xin Zhang; Elke A. Rundensteiner, "Integrating the maintenance and synchronization of data warehouses using a cooperative framework," Information Systems, vol. 27, no.4, pp. 219-243, 2002.

[18]. Yeung, G.C.H.; Gruver, W.A., "Multiagent immediate incremental view maintenance for data warehouses," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol.35, no.2, pp.305-310, 2005

[19]. Yi-Yang Yang; Yain-Whar Si; Wai-Leong Leong, "Data warehousing massive real-time elevator signals and maintenance records," Industrial Technology, 2008. ICIT 2008. IEEE International Conference on, pp.1-8, 2008.

[20]. Xiaogang Zhang; Luming Yang; De Wang, "Incremental view maintenance based on data source compensation in data warehouses," Computer Application and System Modeling (ICCASM), 2010 International Conference on, vol.2, pp.287-291, 2010.

[21]. Yow, T.G.; Grubb, J.W.; Jennings, S.V., "Managing data warehouse metadata using the Web: a Web-based DBA maintenance tool suite," System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on, vol.6, pp.49-54, 1998.

[22]. Lijuan Zhou; Qian Shi; Haijun Geng, "The minimum incremental maintenance of materialized views in data warehouse," Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on, vol.3, pp.220-223, 2010.