I.J. Information Technology and Computer Science, **2016, 9, 9-16**
Published Online September 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.09.02

Developing an Efficient Model for Building Data Warehouse Using Mobile Agent

Tarig Mohammed Ahmed
Department of Computer Sciences, University of Khartoum, Sudan
E-mail: Tarig_harbi71@hotmail.com

Mohammed Hassan Fadhul
Department of Computer Sciences, University of Khartoum, Sudan
E-mail: mfadhul@hotmail.com

***Abstract*—**Data-warehouse is an emerging technology with great potential. Nowadays, businesses are competing fiercely to dominate the market where profitability is promising using every available means to reach their goal. Performance and storage are big challenges in building data-warehouse focusing by researchers recent years. In this paper a new model for developing an efficient data warehouse by using mobile agent technology has been proposed. The main idea behind this model is to use the mobile agent to extract and analyze operational data in their location. So, instead of using ETL, the mobile agent will be used. After the mobile agent completing its journey among operational databases, all tasks of ETL will be performed. By this way no need high storage media to extract the data from the operational database. As cost of time, the model proves less consuming of time. The model has been implemented using .Net Framework and C# and the results have been presented and discussed.

***Index Terms*—**Data-warehouse, Performance, Storage, Mobile Agent.

I. INTRODUCTION

Data-warehouse is an emerging technology with great potential. Nowadays, businesses are competing fiercely to dominate the market where profitability is promising using every available means to reach their goal. “The Data-warehouse is a subject-oriented, integrated, time variant and none-volatile collection of data in support of management decision making process” [1].

Data-warehouse is subject-oriented as the stored data targets specific subject for example, a company’s sales, customers information and similar aggregation semantics which is different than the general data on every day operations[2]. Integrated defining characteristic stresses whether the stored data consolidated from a heterogeneous, it goes through reformatting process to be valid for only one designated schema which suitable for the target database. However, the third defining characteristic, which is time-variant, disregard the age of the stored data as a rule for excluding the data from the loading process or be a reason for pruning. In practice, any data stored in a database has some connection with time somehow, thus, data in a data-warehouse may age between recent to several years old[3]. None-volatile defining characteristic materializes the concept of modification operation on data as not possible after its have been loaded in the data-warehouse. Generally, there are only two operations are allowed to be performed in the data-warehouse including loading data and accessing data [4].

The nature of the business is a major factor in prompting the objectives of the data-warehouses where business constraints decide on the shape of that business. The following points summarize common data-warehouse objectives to most businesses [5].

- To create data repository for a large volume of data that can be used in the analytical business process
- To provide intuitive access to information that will be used in decision making
- To ensure that the appropriate data is available for the appropriate user at the appropriate time
- To leverage information as an asset for the corporation
- To relieve the pressure on the streamline cost
- and increase the return on investment
- To eliminate intensive gathering and report processes
- To build the foundation of fully integrated report platform
- To provide relevant, accurate and timely information to the business

Data-warehouse comprises a set of databases and sub-systems each of which designed to perform a specific task on data. The first component is data source where the Online Transaction Processing OLTP is the center stage and the corporate performs everyday transaction on data such as inserting, modifying and deleting records. [6]

ETL process is a major key in data-warehouse performance. In general, researchers are concerned with the performance of the data-warehouse for quite some times and they still keep proposing new techniques in an attempt to reach the highest possible performance[7]. The problem is challenging in the light of the data-warehouse popularity and the benefits gained, by the organization, out of this technology. Beside performance , the storage is also considered as challenge when building data

Mobile Agent Systems (MAS) is a subarea of distributed systems and it is considered as one of mobile computation trends. The main player of MAS is a mobile agent. The mobile agent is an object that can move autonomously from node to anther in heterogeneous network to achieve tasks on behalf of its owner. The mobility of the mobile agent among nodes (hosts) reduces the network bandwidth because the computation is done in data locations. In addition, many benefits are related to MAS such as reducing the network latency, recovering the communication failure or interruption, etc.[8]

In this paper a model for developing an efficient data warehouse by using mobile agent technology has been proposed. The main idea behind this model is to use the mobile agent to extract and analyze operational data in their location. So, instead of using ETL, the mobile agent will be used. After the mobile agent completing its journey among operational database, all tasks of ETL will be performed. By this way no need high storage media to extract the data from the operational database. As cost of time, the model proves less consuming of time. The model has been implemented using .Net Framework and C#. The paper is organized as following: section 2 presents a survey of the relevant works. Section 3 mentions full description of the proposed model with implementation. Based on results, comprehensive discussion has been conducted in section 4. Finally, conclusion and future work is presented in section 5.

II. RELATED WORK

Many techniques have been proposed by researchers in the area of Data warehouse architecture. The two main issues that are interested by those techniques are performance and storage. Such as following:

There are two approaches to build a data-warehouse each of which follows a different design pattern as well as different performance concerns. The first approach is *Bill Inmon* approach and the other is *Ralhp Kimball* approach. Both approaches obtain data from the data sources but differ in the way they organize data in the Data-warehouse. *Bill Inmon* approach uses data in atomic level (third normal form - *3NF*) style for storing extracted and transformed data. On the other hand, *Ralph Kimball* approach uses multidimensional style (dimension and fact table layout - *star schema*) for storing extracted and transformed data. To fine tune the performance when building a data-warehouse using *Inmon* style, avoid denormalizarion actions which combine or split relation unless it is absolutely necessary [4].

Real-time ETL system has two effects including putting changes of the data source into data-warehouse instead of loading it periodically causing competition for resources between update and query when executed in parallel. To overcome this limitation, researcher *Jie Song* proposed two algorithms, one is a scheduling algorithm and the other is triggering algorithm listener. Those two algorithms were implemented in a framework named *Integrated Base Scheduling Approach* (IBSA) to solve the problems. Balancing is another issue to be handled, which is a mechanism for distributing the resources between concurrent executions of queries. Nevertheless, there are four rules that the proposed scheduling algorithm should consider when deciding whether the next schedule should be query or update. The other part of the framework is balancing algorithm which confined three modes including query mode, update mode and concurrent mode. The three modes together control the organization of the resource usage. According to the result of the experiment, performance is quite stable and high [9].

Jamil, et al. [10] made a comparison between different indexing techniques and observed the impact of variable size with respect to time and space complexity. Some of the indexing techniques advantages which are closely related to high performance. They stated that indexing in data-warehouse is generally a tricky business as it slow down the loading process and need more space if there are too many indices but the query response time is high. On the other hand, if there are few indices then the data load up is fast but the response time is slow. However, the discussed three types of indexing include bitmap indexing, clustering indexing and hash-base indexing, each of which has some advantages that the others have not.

Edgar, et al. [11] proposed a model using the mobility technology to build data-warehouse object. The notion is to process the data locally and the sends the result to the data-warehouse. To understand how this proposed module works and introduce its various components, it should be noted that the agent itself is stored procedure located at the database and realized if the agent dispatcher transferred it to a run-time. Agent run-time environment installed in each data source and based on table and stored procedures. The table holds the table names of the results and a particular state of each agent while the stored procedures designed to execute various tasks related to the agent. Agent dispatcher object, however, is responsible for injecting the agent code-base into database and to provide the required data such as migration option, qualified name of the finishing line table and the result table name so that the run-time environment can execute the agent.

Bhan, et al.[12] developed a tool to measure the data warehouse performance in different environment conditions. The tool has many features to control the critical components and to identify the performance problems. The tool also can justify the performance of OLAP and hardware configuration for different users. As results of using the tool, storage process has been identified as bottleneck. To overcome the this problem, three-tier architecture has been proposed.

Abdulhadi, et al.[13] proposed to summarize the data table to reduce queries response time and a storage space. Specifying the suitable index type is consider as a challenge of this idea. Bitmap and B- Tree indexes have been selected. As results of measuring performance, the bitmap index has a good performance more than the B-tree index.

Moudani et al. [14] suggested to use knowledge discovery process (data mining) for managing data warehouse performance. the main idea of this paper is to use association rules for caching multidimensional queries. This technique has been implemented to measure its impact. But, the data mining process is performance overhead.

Ming-Chuan et al. [15] proposed an algorithm to select properly materialized views in order to enhance the performance. But, the materialized view has a problem with the consistency and the refresh process will be a performance overhead.

III. Proposed Model

The proposed model aims to develop an efficient data warehouse using mobile agent technology. The main idea behind this model is to use the mobile agent to extract and analyze operational data in their location. So, instead of using ETL, the mobile agent will be used. After the mobile agent completing its journey among operational database, all tasks of ETL will be performed. By this way no need high storage media to extract the data from the operational database. As cost of time, the model proves less consuming of time. the model consists of several components, each one has specific role as following:

1. Operational Databases: daily transactions are stored in the operational database.
2. Mobile Agent: the mobile agent plays key functions in the model. The mobile agent visits all operational database to implement all ETL task and carries the results.
3. Agent Home: This component controls in the mobile agents activities by provide them with required tasks to be performed. The mobile agents start and finish their journey in this place.

A. Case study

To verify and test the proposed model to improve data-warehouse performance using mobile agent, we adopted a fictional company that sells electronic appliance and home entertainment equipment. The company has six outlets located over some states of the Sudan including Khartoum state, Nahr El-Neel state and Jazeera state. There are two branches in Khartoum state including Jabra and Safia while Nahr El-Neel has one branch named Atbra and Jazeera state has one branch name Madani. The branch is located in Shmalia province. Each of these outlets sells the company's products to the public and stores the sales transactions in its own database. The database named after the branch name and there is no communication between them. Furthermore, it is assumed that the company rule states that branches should open 24/7 and each branch makes around 700 transactions a day.

To assist the operations for this project, five software have been designed and implemented as follows:

B. Sales Generator

This software is used to generate random transaction and stores it in the branch database. Using this software, over one million and two hundred thousand transactions have been generated for each branch database.

C. Meta Manager

The software is designed specially to configure the Meta database of the data-warehouse where the information about the source and data-warehouse databases are stored in addition to information about loading and refreshing the data-warehouse.

D. ETL

The software is used to extract data from the source database and load it into the warehouse database. In addition, the software writes information of the current operation to the Meta database including information about the elapse time and the affected records in both the source database and data-warehouse database.

E. Agent Manager

This software configures the agent before sending it away to execute the assigned task. It takes information from the meta database and the user configuration values.

F. Agent Home

This is simple designed software which intended to demonstrate working with data-warehouse cube. In addition, it is the place at which the agent comes with the result of the assigned task.

G. Model Implementation

In this section, the implementation of the developed technique will be discussed. Experimenting with developed software is aimed to prove that mobile agent can execute task on behalf of the user on the source data efficiently and with high performance. The developed software is a simulator combining both functions of mobile agent and ETL component which is an important element in the data-warehouse environment. The simulation also addressed a solution for data-warehouse stable storage.

H. Experiment objective

The objective of this experiment is to systematically prove the model of this work in which a claim has been made that using mobile agent to construct OLAP cube task has a better performance than using the traditional methods of ETL for the same task.

I. Experiment Strategy

Test will be conducted using a simulator to simulate the Data-warehouse and ETL process. A simple ETL component operates over both data sources and Data-warehouse in order to process the extraction and loading operation, skipping integration and transformation process as the data sources are homogenous databases.

Practically, there are two phases for the test as follows

- The first phase is to run traditional ETL programme to extract and load data from the data source then construct OLAP cube from that data. The duration of the task will be recorded and stored in the Meta database.
- The second phase is to run mobile agent carrying the same task of constructing OLAP cube on the same data sources and wait for it to get home. The duration of the task will be recorded and stored in the Meta database.

It worth mentioning that the test will be conducted incrementally as there are six data sources. In each incremental round, the elapse time will be recorded to finally asses the new technique scalability.

J. Simulator Design and Implementation

A simulator is designed specifically for the purpose of conducting the experiment that will determine the fate of the model claimed in this paper. Figure 1 shows the architecture of the simulator along with the core components. The architecture shows each component's input and output in term of data direction and the following are the main functions of each in the architecture.

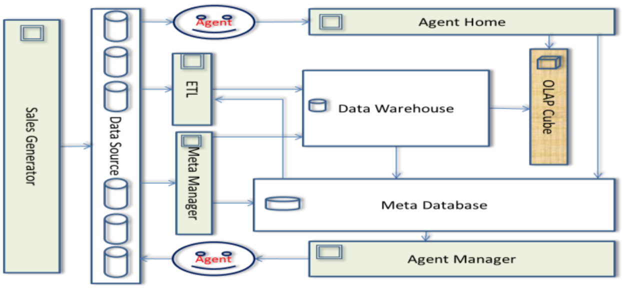


Fig.1. Simulator Architecture

Looking at the architecture, there are two databases confined in the middle namely Data-warehouse and Meta database and data source databases at far left. These are the only databases vaults designed for this simulator. They represent the simulator backend which work together with five user interface components representing the simulator frontend in which users provides configuration values to allow the simulator executes tasks accordingly. The component at the far right which named OLAP cube is in fact stores data in the main memory and it is the final destination of ETL and agent components job.

The design of the data source database is simplified as it is intended for use of this simulator only. It comprises three relations only including the daily sales transaction, items and product type. Figure (2) shows the schema of this database

The second database is the Meta database which is the heart of the simulator and it is used to store information that makes the assigned task of the simulator easy and automated. The schema of Meta database has relations for storing information about data source servers and its contained tables, Data-warehouse relations information, agent and agent's jobs in addition to relations storing ETL and agent activities. Figure (3) is screen shot of Meta database schema and all transactions were stored in this schema

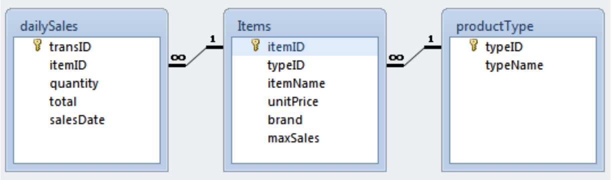


Fig.2. Daily Sales database schema

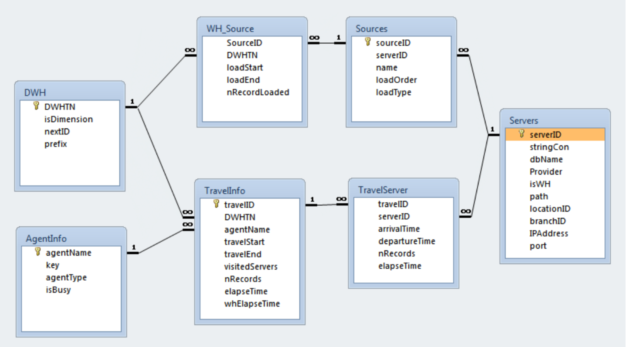


Fig.3. Meta database schema

Finally, warehouse database is the place where ETL application data being processed from the data source. This database is designed following Kimball approach which defines the layout of the star schema. It comprises four dimension tables and one fact table that stores two measures including dollars sold and quantity sold in addition to the foreign keys of the four dimensions. Figure (4) is screen shot for Data-warehouse schema. The data was stored in multi dimensions in this schema.

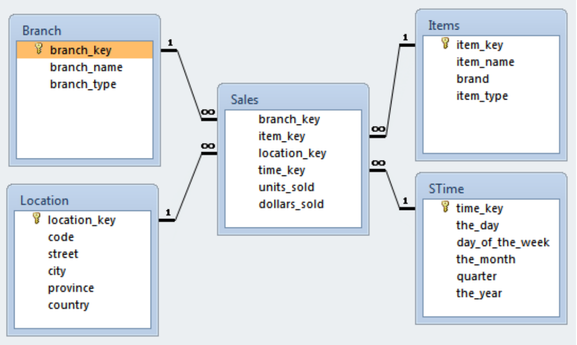
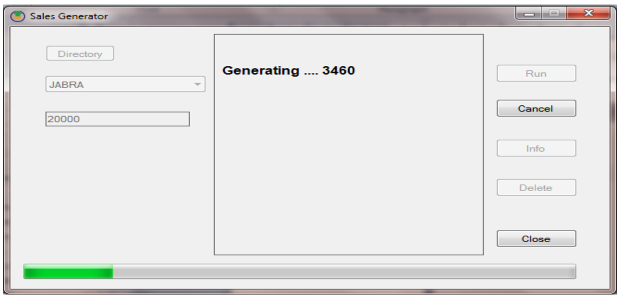


Fig.4. Data-warehouse star schema

K. Sales Generator

This component main function is to generate sales transactions and stores them in the specified data source to imitate customer purchase transaction in one outlet. The generator randomly selects a product, quantity sold and calculates transaction total before it stored in the outlet database. It can generate unlimited number of transactions but it bound to the number specified by the user and the hard disk space. Figure (5) shows Sales Generator tool user interface.



JABRA

20000

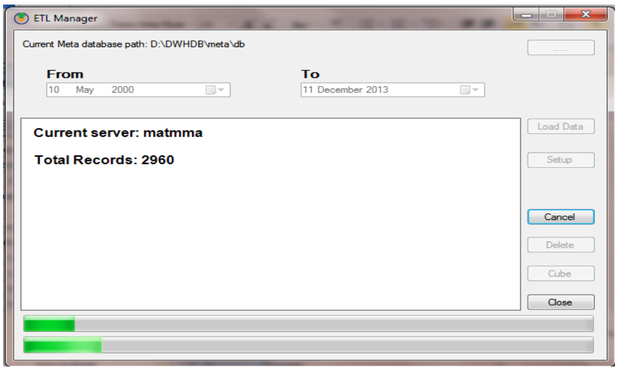
Generating 3460

Fig.5. Screen shot for Sales Generator in action Meta Manager

Meta Manager tool primarily works with Meta database of the simulator to register the data source servers information so it can be used by other applications. It process information about a server such as database name and its tables, provide the server with IP address and port number to be use with mobile agent technology, load order of the table into Data-warehouse to be used by ETL process and some other important information about loading data. It also adds server information to Data-warehouse including branch name and location. Meta Manager is the one that provides information to the simulator for automation.

L. ETL Manager

ETL Manager tool does the hard job of extracting and loading data from the data sources into the data-warehouse on demand. User specifies the period of data to be extracted and the manager does the rest. This operation takes the names of the servers of the data source from the Meta databases and automatically connects to them to extract and load the required data. After loading, the system makes these data ready to be cubed for OLAP process as required. This process achieved by launching Cube constructor user interface from within the manager where the user specifies the type of the cube required and obtains the result. Figure (6) is snapshot of the ETL Manager user interface. This figure mentioned the number of transactions that used in the experiment



Current Meta database path: D:\DWHDB\meta\db

From 10 May 2000

To 11 December 2013

Current server: matmma

Total Records: 2960

Fig.6. ETL Manager Screenshot

Main screen of the application always inform the user about the activity of the process. At the end of the process, information about the result is displayed in details which includes information about the elapse time and the number of records that produced by the operation

The manager allows the termination of the process, if the user which, at any time using the Cancel button. In this case cube will be constructed using only the processed data that have been loaded into warehouse vault. Agent Manager

As shown in the simulator architecture, Agent Manger tunes the agent task in which preparing SQL statement for the agent then create it, initialized it, provide it with all the necessary tools to execute the task and finally send it over the network to do the assignment. When the agent finishes with all servers that have been told to visit, it returns back home carrying the result of the query task and hand it over to be displayed. At home, home manager connects to the Meta database and registers all the information about the agent journey such as departure and arrival time, number of records of the job result and the journey duration.

Constructed cube data do not persist as it and stay in the main memory only. Loosing these data means rerunning the process to display it again and it might different because the agent always brings real-time data.

M. Development machine specification

The test will be conducted on computer machine with the following values

Table 1. Development Machine Specification Table

Name	Value
Machine type	Laptop
Operating system	Windows 7
Processor	Pentium dual-core cpu T4400, 2.20 GHz
System	32 bit
Network	Socket communication
Agent Home port	8085
Data source ports	8086, 8087, 8088, 8089, 8090, 8091
IP address	Local host – 127.0.0.1

N. Environment of the experiment

Output of the experiment consist of a number of values but only elapse time of the process will nr in the interest of the analyses and comparison aspects of both technique. The environment is limited to the develop simulator, using the developed machine and continuous supervision to the behaviour of the running program.

O. Pre-test

Pre-test process intended to fill the databases of the simulator with the necessary data that would trigger the process as required. Thus, all data source have to be registered using Meta manager application. As mentioned above, each outlet in the scenario has its own database; however, the registration of the outlet includes the branch information, the location and specifying IP address along with port number for that branch site. On the other side, the registration has to process storing information in the Meta database about the data-warehouse database including the name, contained tables IP address and the socket port for remote communication. This step is followed by filling the outlet databases with the generated sales transaction using Sales Generator application. Each database filled with over a million transaction records and at max one million two hundred records. Now the simulator has enough information to start the test.

P. Running the experiment

As mentioned above, the test has two phases to complete and having the result on hand. ETL runs on the first phase but this is just an option as it is possible to run the agent first. Both agent and ETL application require selecting a valid Meta data to initial their process, however, in ETL application, the second step after loading Meta data is to specify the data sources servers that ETL process should operate on. This is done by bringing up the Server selection interface by pressing Setup button. When done, user is allowed to shoot ETL process by pressing Load Data button which will finishes the job and display the final statistic of the whole operation. So far the processed data were extracted from the data sources and loaded to target database which the Data-warehouse. but still the test has reach the final destination were the cube is constructed and displayed to the user, however, the user should press Cube button to display cube interface and select the type of the desired cube. By doing so, a final statistic will be drawn beside the cube result containing all information relevant to the cube construction and the elapse time of clean time of the entire operation in seconds. The cube configuration interface displaying the result of constructed cube along with relative information about the operation

The second phase of the test process focused on mobile agent for which an assignment is set and executed to produce the same result as traditional ETL task did. Simply, the agent departs from the agent manager where its construction and configuration of the assignment take place. It then roams all servers that have been told to visit, executing the assigned job and when done, then head home carrying the resulting data of the job and surrender it at home. Home manager is then immediately processes the cube construction upon delivery. When the cube is successfully constructed, the manager displayed along with the relevant statistical data of the agent journey. User can use delete button of the manager to reset meta database to the default values so that the user can have fresh start.

The test process with its two phases applied incrementally up to six servers which is the total of all servers registered in the simulator Meta databases. The result of the test is compiled in the table below

To see the result of the performance of both ETL and mobile agent it been plotted in the graph below

The test has been run twice in order understand the consistency of the behavior of both techniques. The table below shows the compilation values for the same data used in test (1) flowed by figure (7) which is a plotted graph for that compiled data.

Table 2. Compilation of the performance test values

Servers	ETL	Mobile Agent
1	286.200	9.277
2	852.236	15.925
3	1383.272	22.786
4	2007.431	34.794
5	2402.378	37.109
6	3090.000	45.608

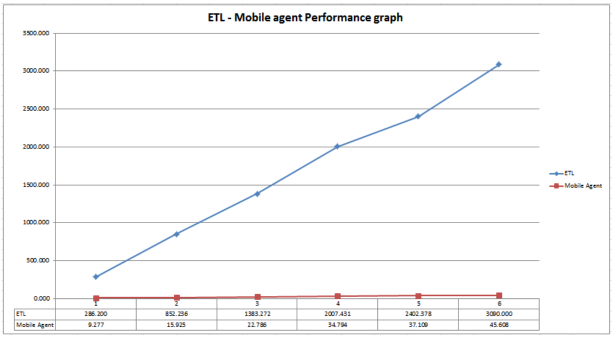


Fig.7. ETL - Mobile Agent performance graph

Q. Result Discussion

The discussion starts with clarifying few points about the test. Even though both techniques end up constructing OLAP cube using the same data records of the data source, they are differ in the way they deal with these data. ETL process loading extracted data into the target database then construct the cube. On the other hand, mobile agent eliminates the process of extracting and loading data into target database. In fact, there is no Data-warehouse or even a target database whatsoever in the mobile agent techniques. The second point worth mentioning is that there is no staging area when applying ETL technique. Staging area is a set of tables mirror those in the Data-warehouse database and they used by ETL process to integrate, clean and reformat data from the data source to match the format of the data-warehouse database before loading process.

The test has been run three times on the same data in an attempt to read the consistency of the result and the variation if any. Indeed, the resulting chart for the different tests proved the variation of the given output. Test one showed a sharp drop round in which the simulator was incremented to process three servers at ETL technique. It also showed that test 2 and three have their plotted graph look almost steady. This behavior is a side effect resulted from the machine CPU usage.

By analyzing the chart of the test, the different is obvious between the two techniques for constructing the resulting cube. It is noticeable at first glance how powerful the agent in processing the cube in term of performance and it can be read easily that the performance of the agent, when constructing a cube out of six data source servers is much far better than constructing a cube with ETL out of one server. However, loading a million of records from remote data source, no doubt is costly in term of time cost which degrades the performance. In addition, the user has to construct the cube as second step from the loaded data ETL but mobile agent does follow this procedure to for construction. The technique took advantage of the mobility feature of the agent and allows user to configure it so that it can run SQL statement which ultimately creates the cube on the remote server and come back with it ready for display. The result records of the SQL statement do not accumulate to more than few tenths or hundred records to build the cube.

To understand the chart talking numeric, simple calculation can do that by comparing the number of time finishing a task by each technique. Consider constructing OLAP cube from one server as a task. When ETL finishes the specified task, mobile agent will finish the same task 3085 time. On the other hand, when mobile agent finished the specified task, ETL will just finish 3.24% of that task. When the task spawns all six servers, the time it takes for ETL to finish once, the mobile agent will finish it 9775 time. Working out all servers will lead to the discussion and measuring the scalability of each technique in order to find out at what level of scalability each technique can hold.

In ETL technique, scalability is much depends on the target machine hardware resource specially the stable storage which could be terabyte or more. On the other hand, mobile agent scalability depends on the available memory of the machine server that is currently working on. This is because the size of the load that carries is increasing whenever finishes with each visited server.

IV. Conclusion and Future Work

A new model has been proposed in this paper to build an efficient data-warehouse. The model aims to improve the performance, it provides real-time data which is one of the concerns for data-warehouse users. However, it can incredibly enhance the time performance for a given data-warehouse tasks. In term of servers to process as resource, mobile agent can freely scale up but other factors may limit the scalability such as available free memory size of the source server.

As limitation of the proposed model, Mobile agent technique discussed and implemented in this work builds the result of the assigned task on the fly, which means on the main memory. The task result scales up when done with each visited server. This behavior jeopardizes the entire operation when the result past available free memory on the server which could deter the agent from completing its assigned task and terminates immediately. If the server that the agent is currently working on crashed for any reason, the agent will terminate there and will never get back home as there is no recovery procedure to follow.

As future work for the developed model, interested researchers can add more feature to it and work on it limitation to improve it further. There are some areas need more attention including complex query task which should aggressively tested as it represents core OLAP and BI power. Algorithm to improve availability is also one area of concern to the new developed technique in which allowing the mobile move on if the resource is not available. Limited memory resource is not in the good of the new technique for which the mobile agent scale up with data whenever finished with visited server, so it is crucial for the live of the agent to find enough room at the memory of the next visited server to avoid jeopardizing the entire mission. To solve the problem an algorithm is needed to deal with the situation. Similarly, passing by a resource server may produce a deceiving task output for the user and hence affect the reliability of the technique. To heal the problem, an algorithm is needed to ensure heterogeneity systems were properly handled by the agent.

References

[1] H. Jiawei, K. Michleline and P. Jiane, “Data Warehouse and Online Analytical Processing 3rd Edition,” in Data Mining Concept and Techniques, Morgan Kaufmann, 2012, p. 126.

[2] Patil, Mayuri Kirange DD. "Use of ETL Subsystems for Real-Time Data-Warehouse using MS SQL Server Tool." Asian Journal of Computer Science and Information Technology 5.4 (2015): 26-32.

[3] Terrizzano, Ignacio, et al. "Data Wrangling: The Challenging Yourney from the Wild to the Lake." CIDR. 2015.

[4] J. L. Wilburt, J. Yang, C. Yiang, G. Hictor and W. Jenifer, “Performace Issue In Incremental Warehouse Maintainence,” IEEE, pp. 461-472, 2000.

[5] A. Thilini and W. Hugh, “Key Orginizational Factors In Data Warehouse Architecture Selection,” ACM, pp. 201-212, 2010.

[6] V. Rainradi, “Introduction To Data Warehouse,” in Building A Data Warehouse With Example in SQL Server, Apress, 2008, p. 10.

[7] Tarig Mohamed Ahmed, Increasing Mobile Agent Performance by Using Free Areas Mechanism.,Journal of Object Technology,6,4,125-140,2007.

[8] Marketos, Gerasimos D. Data warehousing & mining techniques for moving object databases. Diss. Ph. D. dissertation, Department of Informatics, University of Piraeus, 2009.

[9] J. Song, Y. Boa and J. Shi, “A Triggering and Scheduling Approach for ETL In Real-Time Data Warehouse,” IEEE, no. 2010 10th, 2010.

[10] S. Jamil and I. Rashda, “Performance Analysis of Indexing technique In Data Warehousing,” IEEE, pp. 57-61, 2009.

[11] Weippl, E.; Altmann, J.; Essmayr, W. Mobile database agents for building data warehouses Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on, Pages: 477 – 481, 2000

[12] Bhan, M., Rajinikanth, K., Geetha, D. E., & S Kumar, ,T.V. (2014). DWPPT: Data warehouse performance prediction tool. International Journal of Computer Applications, 104(13)

[13] Abdulhadi, Z. Q., Zuping, Z., & Housien, H. I. (2013). Bitmap index as effective Indexing for low cardinality column in data warehouse. International Journal of Computer Applications, 68(24)

[14] Moudani, W., Hussein, M., Moukhtar, M., & Mora-Camino, l. (2011). Enhancement of a data warehouse performance using association rules technique. International Journal of Computer Applications, 21(7)

[15] Ming-Chuan Hung, Man-Lin Huang, Don-Lin Yang, Nien-Lin Hsueh, Efficient approaches for materialized views selection in a data warehouse, Information Sciences, Volume 177, Issue 6, 15 March 2007, Pages 1333-1348, ISSN 0020-0255

Authors' Profiles

Tarig M. Ahmed is an associate Professor at Dept. Computer sciences, University of Khartoum. He holds Ph.D. degree in computer science. His main interested research areas: Mobile agent system architecture, mobile agent security, distributed systems, database Systems and artificial intelligent systems. Tarig is an Oracle Certified professional (OCP) from Oracle company as application developer 6i. Also, he has used.Net technology, C#, Java languages

Mohammed Hassan Fadhul was born 29/06/1960, he is a well-organized, reliable and enthusiastic computer science researcher with mature knowledge and practice of software analysis and programming, based on the scientific standards. Some of my strength is the ability to learn new skills and a real commitment to achieving my personal and professional goals within a preset time frame. My journey with the computer science field started in the UK where I obtained a diploma in computer science, form Lambeth College, in 1998. This achievement was followed by obtaining a bachelor from Kingston University in 2005, in the same field. Lastly I have been granted, in 2014, a master degree in computer science from Khartoum University.

How to cite this paper: Tarig Mohammed Ahmed, Mohammed Hassan Fadhul, "Developing an Efficient Model for Building Data Warehouse Using Mobile Agent", International Journal of Information Technology and Computer Science (IJITCS), Vol.8, No.9, pp.9-16, 2016. DOI: 10.5815/ijitcs.2016.09.02