Egyptian Informatics Journal (2016) 17, 147–159

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

FULL-LENGTH ARTICLE

A novel approach for intelligent distribution of data warehouses

Abhay Kumar Agarwal [*], **Neelendra Badal**

Kamla Nehru Institute of Technology, Sultanpur, U.P., India

Received 17 March 2015; revised 23 September 2015; accepted 19 October 2015
Available online 28 November 2015

KEYWORDS

Data warehouse;
Distributed;
Hierarchy;
Intelligent;
Meta Data

Abstract With the continuous growth in the amount of data, data storage systems have come a long way from flat files systems to RDBMS, Data Warehousing (DW) and Distributed Data Warehousing systems. This paper proposes a new distributed data warehouse model. The model is built on a novel approach, for the intelligent distribution of data warehouse. Overall the model is named as Intelligent and Distributed Data Warehouse (IDDW). The proposed model has N-levels and is based on top-down hierarchical design approach of building distributed data warehouse. The building process of IDDW starts with the identification of various locations where DW may be built. Initially, a single location is considered at top-most level of IDDW where DW is built. Thereafter, DW at any other location of any level may be built. A method, to transfer concerned data from any upper level DW to concerned lower level DW, is also presented in the paper. The paper also presents IDDW modeling, its architecture based on modeling, the internal organization of IDDW via which all the operations within IDDW are performed.

1. Introduction

The basic definition of DW was given by Inmonin [10] that DW is a subject-oriented, integrated, time-varying and non-volatile collection of data in support of the management's decision-making process. Generally, an organization starts with the centralized DW system. This centralized DW is responsible, for storing the entire data of the organization, answering all the queries and for decision making. For an organization, as the amount of data and number of queries to be answered increases to an extent, the need for the distribution of stored data generates, that in turns develops the need of Distributed DW. The need for Distributed DW is also developed when an organization grows in size with increase in number of its branches or with the increase in the size of its branches. Many approaches were proposed by many researchers for the distribution of data and construction of Distributed DW. One such approach is distribution of data in hierarchal fashion called Hierarchal Distributed DW. For building Hierarchal Distributed DW either of two design approaches, Top-Down or Bottom-Up is used. Bottom-up Approach

[*] Corresponding author.
E-mail addresses: abhay.knit08@gmail.com (A.K. Agarwal), n_badal@hotmail.com (N. Badal).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

http://dx.doi.org/10.1016/j.eij.2015.10.002
1110-8665 © 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

[18,3,7]: is suitable when the objective of the design is to integrate existing database systems. The bottom-up design starts from the individual local conceptual schemas and the objective of the process is integrating local schemas into the global conceptual schema. Top-Down Design [8,17,16]: In the top-down design approach the data warehouse is built first. The data marts are then created from the data warehouse. Top-Down design approach is generally used for a very large system. An example of Top-Down design approach is DW at country level is distributed among the DW's at state level, and each DW at state level is distributed among DW's at city level and so on up to the local mart level. The advantages of doing so are local queries are answered locally while global queries are answered by the system as a whole. The IDDW proposed in this paper uses this very approach for building it.

The steps followed in designing and building IDDW are as follows: (1) Selecting a theme, (2) Framing, a theme in hierarchal structure of N-levels, (3) Identification of N-Levels in hierarchal structure, (4) Identification of locations in each levels of hierarchal structure, (5) Formation of hierarchal structure considering the identified locations, (6) Initial, location and number of DW's to start within the formed hierarchal structure, (7) Data to be stored in each DW's, (8) Incorporating any new DW build at any location, (9) Transfer of the data/hierarchal information, related to a location, in the new DW build at that location and (10) Stores the local data of a location in the DW if available at that location. The detail is presented in Section 3 of the paper.

The other sections in the paper are as follows: Section 2 presents the Related work; Section 3 presents the proposed work that includes the IDDW Modeling; the IDDW architecture; IDDW data model; the internal organization of IDDW which comprises of global MDS, local MDS etc. and with the help of which all the operations in IDDW are executed; the construction of N-level hierarchal structure, IDDW using three algorithms; in Section 4, A case study is taken for IDDW, showing an education system in India in 8 level hierarchal structure; and in Section 5, Experimental setup is developed, based on an 8-level hierarchal structure. A program was written in JAVA with SQL in backend. The real-time data is inputted using a front page in the system and is intelligently stored in the concerned DW in the system, and in Section 6, various observations and analysis are made about IDDW based on experiment performed.

2. Related works

The Distributed DW facilitates the policy and decision makers, by providing a coherent and single view of data. It does so inspite of the fact that data are physically distributed across multiple DW's in multiple systems at different branches. Many authors proposed many techniques for developing Distributed DW.

Inmon in [9] proposed an approach of building the distributed data warehouse. The approach assumes the presence of both local and global data warehouses with data stored in each are mutually exclusive. The local DW includes local data of interest, while, the global DW contains common data across the organization and data integrated from various local DW. Inmon's assumption, about the mutual exclusivity of data between the local and global data warehouses, seems to be impractical. White in [22] proposed an approach which is named as "Two Tier Data Warehouse" and is a combination of centralized data warehouse and a decentralized data marts. Noaman et. al. in [15,14] proposed an architecture for Distributed DW. It uses Top-Down design approach and presents two fundamental issues: fragmentation and allocation of the fragment to various sites. Work proposed by author in [15] is extended work in [14]. They extend it by describing the functionality of distributed DW system architecture components, by giving definition of relational data model for DW and by giving a horizontal fragmentation algorithm. Zhou et. al. in [24] proposed hierarchically distributed data warehouse (HDDW). HDDW integrates the local data marts into a hierarchy. HDDW uses bottom-up design approach for building the warehouse. This method is useless in case of having a large central data warehouse needed to be a distributed data warehouse (top-down approach) because no fragmentation schema has been used. Bernardino et al. in [2] designed a new technique called data warehouse stripping (DWS), which is a round robin data partitioning approach for relational data warehouse. The limitation of DWS technique is that it is not effective in data warehouse with big dimensions. Therefore, the researchers propose a new approach called selective loading to deal with data warehouses with big dimensions in DWS systems. The selective load technique proposed explores the fact that the subset of the fact table rows stored in each node is only related to a small part of the rows of the big dimension and not related to all of them. Thus, the idea is to store in each node only the dimension rows that are related to fact rows stored in that node, and not to replicate the entire dimension [5]. This method faces a problem because the data size will be bigger over time and the round-robin partitioning approach must be applied again to partitioning the fact and dimension table row-by-row through all nodes.

Su et al. in [19] proposed a new model of Radio Frequency Identification (RFID) distributed data warehouse, named as RFID-CHDDW. As the name suggests Concept Hierarchy Distributed Data Warehouse (CHDDW) the constructional approach for making it is hierarchical, bottom-up based on the concept hierarchy, it combines the qualities of both data marts and distributed data warehouse.

Grid based approach for distributed data warehouse is proposed in [20,6,13,21]. The other methods proposed by various authors are [23] used ASM to design distributed data warehouse, [4] proposed a new analytical model using Petri Net for distributed data management in a data warehouse and [12] presented and proposed a data warehouse decentralization strategy based on cost-based fragment allocation and replication algorithm.

The IDDW proposed in this paper uses top-down design approach. However, it has advantages over other Distributed DW using top-down design approach. The advantages that the IDDW provides over others are as follows: first, that there is no need to place a DW at each location, rather a DW can be built anytime at any location as per the need of that location; second, IDDW architecture is such that every user registered in the system (i.e. data of registered user stored in a DW within IDDW) is uniquely identified; third, operations to be performed on the system are not location specific rather it can be executed from anywhere within the system; fourth, the internal organization of the proposed system is such that it makes possible for transfer of data from one DW to another.

3. Proposed work (IDDW)

The process to build IDDW starts with the selection of a theme. The theme selected should be such that it may be broken into sub-themes. Each such sub-theme should further be broken into sub-themes and so on.

Once the proper theme is selected, the next step is framing of theme in hierarchal structure of N levels. Framing is done keeping in mind two points: first, on reading any hierarchy in hierarchal structure from top to bottom one can identify any user as a unique; second, and a DW can be build as per requirement at any location of any level at any stage.

After the selection of proper theme and it's framing, the N levels of hierarchal structure are identified. As the levels are identified the various locations in each level are also identified. By doing so, the various levels and different locations in each level are known and it helps in the formation of the overall N-level hierarchal structure.

The formation of the overall structure starts by building a DW at only location of top most level of hierarchal structure. Presently, this DW acts as a centralized DW and stores all the records (i.e. data local to all the location and overall data related to Common table) related to all the levels until any other DW is formed at some other location of any other level. Any DW in the structure stores three types of data: first, is the data that uniquely identify the user in the system; second is the data local to the location; and third is the Meta data. A Common table is used in each DW for storing first type of data while the Other tables are used for storing second type of data. The Common table is also used for the incorporation of any new DW build at some other location of the structure arranged in star schema in each DW. It does so by transferring of the data (records) in the Common table of newly build DW from the DW available at next higher level in its hierarchy. The content of Common table is made so in such a way so it is able to perform all the desired tasks.

3.1. The IDDW modeling

In this section, we propose IDDW modeling. The proposed modeling helps to number a DW as and when built at any location in the IDDW. The number given to any DW, 'D' using the proposed modeling is $D^L_{J_L}$ where 'L' the superscript represents the level number of the location where DW has built and subscript 'J_L' tells about the hierarchy (from top) of the location where DW has been built.

The proposed modeling uses three equations numbered 1, 2 and 3. The Eq. (1) numbers the DW as and when builds at any location in the IDDW. $D^L_{J_L}$ Used in Eq. (1) is obtained by recursively calculating J_L from Eqs. (2) and (3). The three equations are as follows:

$$\mathrm{IDDW} = \{D^L_{J_L}\} \tag{1}$$

$$\{J_L\} = \{J_{L-1^I}\} \tag{2}$$

$$J_0 = \mathrm{null} \tag{3}$$

where

D is a data warehouse.

L is level number i.e. $\{1, 2, 3, \ldots, N\}$ with 1 as top most level, 2 as second level from top and so on up to N.

I is an integer variable and varies from one to K_L.

Here, K_L represents the pre-identified number of locations where DW's may be built at any particular level L in a hierarchy.

J_L is an integer depending on the value of L and calculated recursively using Eqs. (2) and (3).

$D^L_{J_L}$ represents, the data warehouse number that is built, at some location in particular level L.

Based on the proposed modeling for IDDW, using Eqs. (1–3) general architecture of IDDW can be formed. The detail about the formation and explanation of architecture is provided in next section.

3.2. The IDDW architecture

In this section we present general IDDW architecture. Detailed explanation regarding architecture is also presented in this section. The IDDW architecture, based on the modeling proposed in previous section, is shown in Fig. 1. It is a hierarchal in nature and has N levels. Each level is marked from top as Level 1, Level 2 and so on up to Level N as last level. Each Level L in IDDW architecture comprises of K_L number of pre-identified locations. The number of levels and number of locations in each level are identified on the basis of theme selected, for which IDDW is to be formed. The IDDW architecture starts by building a DW at the only location in Level 1. Therefore, we get $L = 1$ and by using Eqs. (2) and (3) we get $J_L = 1$. Thus by using Eq. (1) the number assigned to this DW is D^1_1. The number D^1_1 signifies us that one DW has been built at Level 1.

As per the theme selected and its framing, there are K_2 numbers of locations at Level 2. These are the locations where DW can be build as per the need of particular location. Any DW built at any of the location at this level is numbered, again by using Eqs. (1–3). As it is Level 2, we get $L = 2$ and the set of values of J_L as {11, 12, 13 ... 1 K_2}. The Fig. 1 shows that at Level 2, 'K_2' numbers of DW's are built. Thus each of K_2 numbers of DW's is numbered from D^2_{11} to $D^2_{1k_2}$. Any such number, for example $D^2_{1k_2}$, signifies that DW built at k_2th location belongs to Level 2. It also signifies that DW is in hierarchy under location with DW D^1_1 available at Level 1.

As per theme selected and its framing, there are K_3 numbers of locations at Level 3 for each location at Level 2. The DW may be built at any of these locations as per need. The value of K_3 may vary for each location in Level 2. The DW's formed at this level are numbered on the similar basis as done at Level 2. Any such DW numbered at Level 3 is $D^3_{1k_2k_3}$. This number tells that DW is at k_3^{th} location of Level 3 and it is in hierarchy under location with DW $D^2_{1k_2}$ at Level 2 and DW D^1_1 at Level 1.

The similar procedure is followed for all the levels in IDDW, top to bottom up to $N-1$ level.

For achieving uniqueness for every user registered in IDDW, last level i.e. Level N of IDDW is build differently. A unique three digit ID is generated serially by executing designed predefined method of generating it. This ID is unique for all the K_N location under each location of Level $N-1$. Any two users in same hierarchy, have all the locations same from top to bottom up to Level $N-1$, thus have all the fields

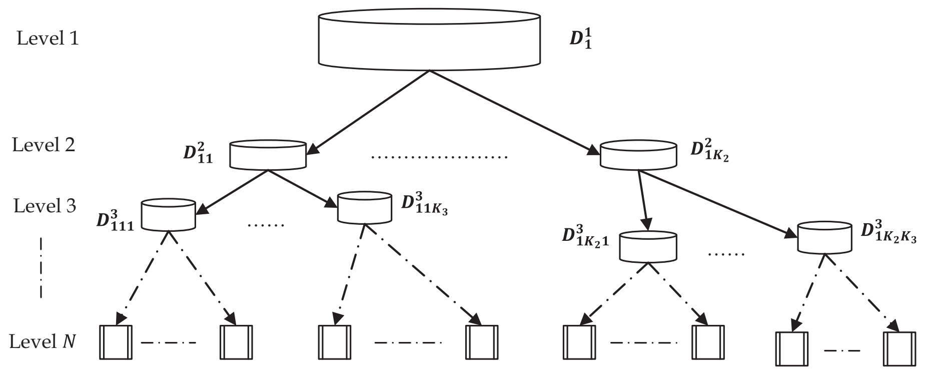


Figure 1 The IDDW architecture.

common. The unique three digits ID generated at Level N differentiate the two.

For storing, the hierarchal information about each location and the information local to a location, in the DW available at that location, each DW in IDDW needs a proper data model. The data model of IDDW is explained in detail in the next section.

3.3. IDDW data model

In this section, we give detail description about the data model used in IDDW. Each DW, whenever built at any location contains three types of tables: first, a Common table; second, Other tables required by particular location and third, a table that stores a Meta data.

3.3.1. The Common table

The Common table that is created in each DW is shown in Fig. 2. The numbers of column, their names and their types are same in each DW and depend upon the number of levels in IDDW. For the N-level architecture there will be N columns in the Common table. The name of each column in Common table is same as that of the name of each level, respectively. The column with name Level 1 stores the locations in top most level, while column with name Level 2 stores the locations in second from top most level of IDDW and so on up to Level $N-1$. A unique three digit ID is stored in the column with name Level N.

Overall the Common table stores the data that provides the name of all locations, used in a hierarchy, level-wise from top to bottom up to the location of level a user belongs to and a unique three digits ID that is serially generated for the every user belonging to that hierarchy.

For example if a user belongs to some location in Level A (where $A \leqslant N-1$) and then for this user the row in Common table contains the following: the name of locations only up to Level A; the null value in other fields up to Level $N-1$; and unique three digits ID in the field of Level N. This information is stored in the Common table of DW, if available at that particular location of Level A. However, in case of non-availability of DW at the particular location of Level A, then the information is stored in the DW available at the next location of the level, higher to level of this location.

Level 1	Level 2	⋯	Level A	⋯	Level $N-1$	Level N

Figure 2 The Common table.

3.3.2. The Other tables

The Other tables in each DW vary as per the requirement of the particular location. For a location the numbers of column, their names and their types in each of Other tables of DW vary. Each such Other tables in a particular DW are connected using the Star Schema of building the DW because of its advantages as discussed in [11]. The Star Schema includes one Fact table that stores the primary key in each of Other tables and as many Dimension tables as the numbers of Other tables. Snowflake schema may also we used in place of Star Schema, but the snowflake schema is slightly more complicated than the Star schema [1].

The data that is stored in the Other tables, provides the information, that is local to a location and is required for decision making at local level.

3.3.3. Table for storing Meta Data

Meta data is the data about data. For the storage of it a table is created in each DW.

To understand further the role of Common table, Other tables and Meta data in each DW builds, let us consider the following scenarios.

Initially, with only one DW, D_1^1 at Level 1 in IDDW. The common table of DW, D_1^1 stores the records of all the users, registering in IDDW. Other tables of DW, D_1^1 store the information local to this location. The table created to store Meta data stores all the information about the records in Common table and about the records in Other tables of DW, D_1^1.

Scenario 1: DW is formed in the locations of two consecutive levels one after the other, say at Level 2 after at Level 1. A new DW says 'D' is build at some location, in Level 2 numbered as D_{11}^2. All the records, in the Common table of DW D_1^1, with value in the first field of a Common table having the name of location of D_1^1 and with value in the second field of a Common table the name of location of D_{11}^2 are transferred to DW D_{11}^2. The Other tables of this newly constructed DW D_{11}^2 now store the information local to this location. The table created to store Meta data of DW, D^2_{11}, stores all the information, about the records in its Common table and about the records in its Other tables. Also the Meta data of DW, D^1_1 is updated as well.

Scenario 2: DW is formed in the locations of two non-consecutive levels one after the other, say at Level 3 only at after Level 1. A new DW says '*D*' is build at some location, in Level 3 numbered as D^3_{121}. All the records in Common table of DW D^1_1 are transferred to DW D^3_{121} with the following values in: (1) first field of a Common table containing the name of location of D^1_1 (2) second field of a Common table containing the name of location, in Level 2, which is in hierarchy with location of D^3_{121} and (3) third field of a Common table containing the name of location of D^3_{121}. The Other tables of this newly constructed DW, D^3_{121} store the information local to this location. The table created to store Meta data i.e. the information, about the records in Common table and about the records in Other tables of DW, D^3_{121}. Also the Meta data of DW, D^1_1 is updated.

The incorporation of a new DW built at some location of any level needs the transfer of records from the Common table of one DW to the Common table of newly build DW, storing of local information in Other table of newly build DW and updation of Metadata in each DW. An internal organization that manages the necessary actions in IDDW is presented in the next section.

3.4. Internal organization of IDDW

In this section, we present the internal organization in IDDW. The internal organization in IDDW is shown in Fig. 3. It comprises of one Local Monitoring and Discovery Services (Local MDS) for each DW build at any location of each level, a Global MDS, a Transaction Manager between the two levels and Discovery Service to connect Global MDS to each of Local MDS. On, in all IDDW is able to perform three operations: first, transferring of records from top to bottom, in a hierarchy, from one DW to another DW; second, storing of data in each DW; and third, updating of information in each DW.

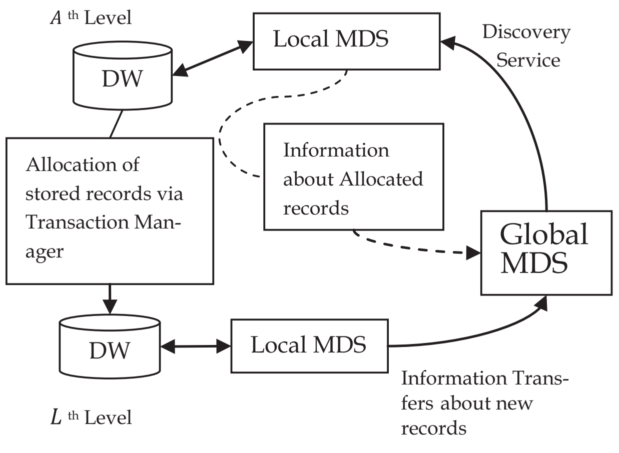

A th Level
DW
Local MDS
Discovery Service
Allocation of stored records via Transaction Manager
Information about Allocated records
Global MDS
DW
Local MDS
L th Level
Information Transfers about new records

Figure 3 Internal organization of IDDW.

The work of Local MDS is to keep track of the records stored in the DW. Global MDS keep track about every record in the IDDW. Global MDS do it so with the help of various Local MDS that send's, record, information to Global MDS. For the explanation of the internal organization in IDDW, we considered two locations in two different levels of IDDW. The two locations considered are from same hierarchy. One location is from Level *A*. The other location is in Level *L*. The Level *A* is always higher to Level *L*. Currently a DW existing at a location in Level *A*, stores, in the Common table of it all the records concerned to this location as well as all the records concerned to all the locations under it in a hierarchy.

Whenever a new DW is built, at the location in Level *L* then the records in Common table of DW at Level *A* are transferred to Common table of newly built DW at the location in Level *L*. The records that are transferred have same, name of locations, as in hierarchy, from Level *A* to Level *L* in the corresponding fields of Common table. The transfer of records from one DW to another is performed via Transaction Manager. At this stage Local MDS of Level *A*, DW, transfers the information of allocated records to Global MDS. The content of Global MDS gets updated by the obtained information from Local MDS. As new user that belongs to location of Level *L*, where a new DW is built, gets registered their records are stored in the DW and its information is passed to Global MDS via its Local MDS. Any DW at the location at higher level may access this information from Global MDS via Discovery services.

Once the IDDW modeling is done, the data model in each DW is known and internal organization of IDDW is designed and the next is the construction of IDDW. In the next section the procedure of IDDW construction is presented. The procedure explains the construction of different levels of IDDW. The three algorithms are presented for the construction of different levels in IDDW. Algorithm 1 is used for the construction of Level 1. Level 2 to Level $N-1$ of IDDW is constructed using, Algorithms 2 and 3. The choice of algorithm between 2 and 3 for the construction of IDDW depends upon the two cases discussed in the section. For the construction of Level *N* a separate procedure is given.

3.5. Construction of IDDW

In this section, we explain the construction process of IDDW. The construction process always starts with the building of a DW at Level 1 and then moves to other levels from top to bottom. Algorithm 1 is used to build a DW at a location in Level 1. Once the DW is build at a location in Level 1, for a hierarchy from top to bottom, two cases can occur: Case 1: The DW can be build at a location in immediate lower level say Level 2, than at a location in Level 3 and so on up to Level $N-1$; and Case 2: DW is formed in the locations of two non-consecutive levels one after the other (for example at Level 3 or Level 4 only at after Level 1), for now say at Level 3. For the case 1, Algorithm 2 is used to build DW at various levels and for case 2, Algorithm 3 is used. The terminology that is used in three algorithms is as given below:

$M^L_{J_L}$ Represents the Meta data of a DW,$D^L_{J_L}$.

$R^L_{J_L}$ Represent the set of records in Other tables of corresponding DW, $D^L_{J_L}$.

R Comprises of the set of records in Common table of a DW (D_1^1).

3.5.1. Build a DW at Level 1

To build a DW at a location in Level 1, Algorithm 1 is used, which is as follows:

Algorithm 1 (*Build a DW at a location in Level 1, following the steps given below:*).

(a) Label the DW as (D_1^1) using Eqs. (1–3).
(b) In D_1^1, create the Common table shown in Fig. 2 and Other tables as required.
(c) Populate; the Other tables with set of records R_1^1 and Common table with set of records *R*.
(d) Create the table for storing the MetaData (M_1^1) of the DW.

3.5.1.1. Building of a DW for case 1. To build a DW at a location in immediate lower level say Level 2, Algorithm 2 is used, which is as follows:

Algorithm 2 (*Build a DW at a location in Level 2, following the steps given below:*).

(a) Label the DW like ($D_{11}^2, D_{12}^2, D_{13}^2 \ldots D_{1K_L}^2$) using Eqs. (1–3) say D_{11}^2.
(b) In D_{11}^2, create the Common table as shown in Fig. 2 and Other tables as required.
(c) Create the table for storing the Meta Data (M_{11}^2) of the DW.
(d) Aggregate set of records *R*, in DW D_1^1.
(e) From the aggregated set of records *R* in DW D_1^1, all the records in Common table of DW D_1^1 are distributed to DW D_{11}^2 having the following values: (1) first field of a Common table contains the name of location of D_1^1, (2) second field of a Common table contains the name of location of D_{11}^2. Keeping the information of transfer records in its table for storing the Meta Data.
(f) The Common tables of the DW, D_{11}^2 is populated with the set of records obtained from DW D_1^1. The Common table of D_1^1 is left with set of records *R* minus records distributed.
(g) Populate; the Other tables of DW D_{11}^2 with the set of records R_{11}^2.
(h) Each time a new record is stored in the Common table of a DW D_{11}^2 its Meta data is updated. Subsequently, its information is passed to data warehouse with which it is connected from Level 1.

Likewise, any other DW can be built at Level 2, for each DW built at Level 2 a DW can be built at any location in Level 3 and so on up to Level $N-1$ one after the other.

3.5.1.2. Building of a DW for case 2. To build a DW in the locations of two non-consecutive levels one after the other, say at Level 3 only at after Level 1, Algorithm 3 is used which is as follows:

Algorithm 3 (*Build a DW at a location in Level 3, following the steps given below:*).

(a) Label the DW in Level 3 using Eq. (1–3) say D_{121}^3. The labeling is done keeping in mind the hierarchy from a location in Level 1, to the location in Level 2 and to the location in Level 3 where DW is built.
(b) In D_{121}^3, create the Common table as shown in Fig. 2 and Other tables as required.
(c) Create the table for storing the Meta Data (M_{121}^3) of the DW.
(d) Aggregate set of records *R*, in DWD_1^1.
(e) From the aggregated set of records *R* in DWD_1^1, all the records in Common table of DWD_1^1 are distributed to DW D_{121}^3 having the following values: (1) first field of a Common table contains the name of location of D_1^1 (2) second field of a Common table contains the name of location, in Level 2, which is in hierarchy with location of D_{121}^3 (3) third field of a Common table contains the name of location of D_{121}^3. Keeping the information of transfer records in its table for storing the Meta Data.
(f) The Common tables of the DW, D_{121}^3 is populated with the set of records obtained from step (e). The Common table of D_1^1 is left with set of records *R* minus records transferred.
(g) Populate; the Other tables of DW D_{121}^3 with the set of records R_{121}^3.
(h) Each time a new record is stored in the Common table of a DW D_{121}^3 its Meta data is updated. Subsequently, its information is passed to data warehouse with which it is connected from Level 1.

3.5.2. Construction of Level N

The last level, i.e. Level *N*, does not build any data warehouse at any of its location. Rather, a process is executed at this level to generate a unique three digits number. The number is generated serially at each location in this level. For one location of Level $N-1$, 999 unique numbers (from 001 to 999) are generated. The three digit number acts as a unique Id for a user. The unique Id helps to distinguish any two users in a hierarchy even if all the locations of a user are same from Level 1 to Level $N-1$.

In the next section a case study is taken in which an 8-level hierarchal architecture analogous to IDDW is developed. The theme taken in case study for which 8 levels are framed is "higher education in India". The data model included in the every DW builds; within the hierarchal architecture is also presented in the case study. How, various levels of hierarchal architecture are constructed in case study is also presented.

4. Case study

India is a country that comprises of many states. Each state has several universities in it. Each of these universities has various institutions affiliated to them. Each institute has many departments running in it. In each department there are various categories of peoples (such as faculty, non-teaching staff, peons, students). Each category of people has various positions (like in faculty one could be either professor, associate professor or an assistant professor). If overall is considered, year-wise, a hierarchical structure can be formed with year at top level and position as lowest level. Still if one more level is introduced, in hierarchy as lower most level, which generates a unique number for every person registered in the system, than each person can be uniquely identified within the system. So this level is included in the hierarchical structure. For example in a year 2014 a state Uttar Pradesh has a university named GBTU. Among the various institutes affiliated to GBTU one is KNIT. There are many departments associated with KNIT one of which is CSE. The CSE department includes many possible categories, one of which is faculty. The category faculty includes many possible positions, one of which is a professor. As there could be many professors each is assigned an identity which is a unique three digits number and is serially generated.

Thus overall framework is fitted to form an 8-level hierarchal structure as shown in Fig. 4. Each, level of 8- level hierarchal structure is named accordingly from top to bottom as follows: Level 1 is named as Year; Level 2 is named as State; Level 3 is named as University; Level 4 is named as Institute; Level 5 is named as Department; Level 6 is named as Category; Level 7 is named as Position; and Level 8 is named as Identity.

The Fig. 4 shows, a year numbered as *Y*: 10 in Level Year. For a year numbered as *Y*: 10 there are 100 locations, numbered from *S*: 00 to *S*: 99 in Level State. For each location in Level State there are 100 locations numbered from *U*: 00 to *U*: 99 in Level University. For each location in University there are 1000 locations numbered from *I*: 000 to *I*: 999 in Level Institute. For each location in Institute there are 100 locations numbered from *D*: 00 to *D*: 99 in Level Department. For each location in Department there are 10 locations numbered from *C*: 0 to *C*: 9 in Level Category. For each location in Category there are 10 locations numbered from *P*: 0 to *P*: 9 in Level Positions. For each location in Position 999 unique three digit numbers (ID) are generated, in Level Identity.

The value of K_L, defined in Section 3 of this paper, is the pre-identified number of locations where DW's may be built, at any particular level *L*, in a hierarchy. From Fig. 4, one can find out the different value of K_L for each level. The obtained values are shown in Table 1. Hence, the total numbers of hierarchies that can be inferred, from 8- levels hierarchal structure shown in Fig. 4, are of the order of 10^{14}. Out of so many hierarchies one is shown in blue rectangular box in Fig. 4.

A graph is plotted using the value shown in Table 1. The graph plotted is shown in Fig. 5. In a graph, Level Number is considered at *x*-axis and the value of K_L, in power of 10, at *y*-axis.

It can be seen, for the case study shown in Fig. 4, from the Table 1 and a graph shown in Fig. 5 that 10^{14} different user registered in the system gets unique ID. The number of unique ID's is same as number of hierarchies i.e. 10^{14}. One can also deduce from the graph the maximum number of different users, gets registered, belonging to a particular level.

4.1. The data model used in case study

For a case study presented, an 8-level hierarchical structure is formed. A, DW may be built at any location of any level in this an 8-level hierarchical structure. A, DW that built includes the same data model as discussed in Section 5. Therefore each DW build includes the following:

4.1.1. Common table

As per the structure of Common table discussed in Section 5, number of columns in the table should be same as that of number of levels in hierarchical structure. Also the name of each column must be the same as that of the name of each level

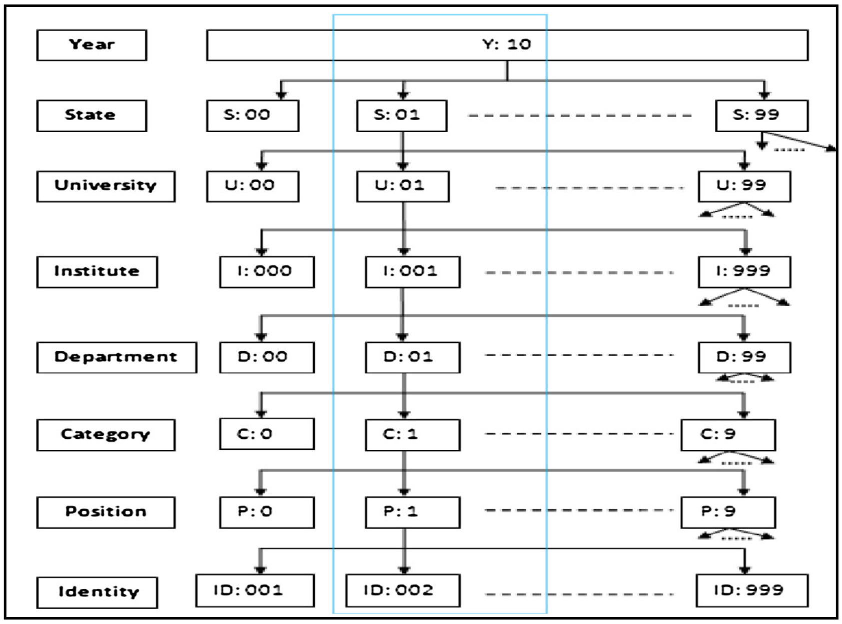


Figure 4 A case study representing 8-level hierarchal structure.

Table 1 Number of Hierarchies generated up to each level number as per the value of K_L.

Level No. L	Name of level	The value of K_L for each level L	No. of hierarchy up to level L
1	Year	$K_1 = 1$	1
2	State	For each K_1 number of $K_2 = 10^2$	10^2
3	University	For each K_2 number of $K_3 = 10^2$	10^4
4	Institute	For each K_3 number of $K_4 = 10^3$	10^7
5	Department	For each K_4 number of $K_5 = 10^2$	10^9
6	Category	For each K_5 number of $K_6 = 10^1$	10^{10}
7	Position	For each K_6 number of $K_7 = 10^1$	10^{11}
8	Identity	For each K_7 number of $K_8 = 10^3$	10^{14}

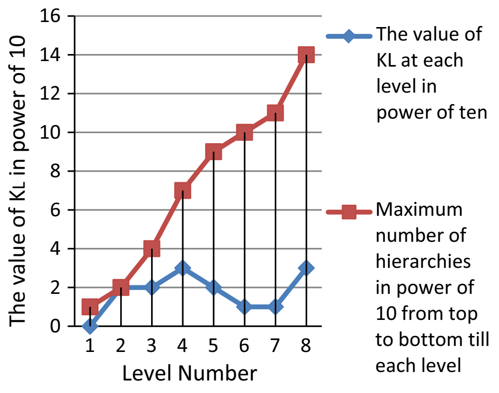


Figure 5 Hierarchies generated up to each level number, as per the value of K_L.

respectively. As there are 8-levels in the hierarchical structure formed and shown in Fig. 4, so there are 8 columns in each Common table. Also the name of each column in Common table is same as that of each level respectively as shown in Fig. 6.

4.1.2. Other tables

The Other tables included in each DW build, at any location of any level in an 8-levels hierarchical structure, are two. The number of fields in each of these tables in every DW may vary and depends upon the requirement of the location.

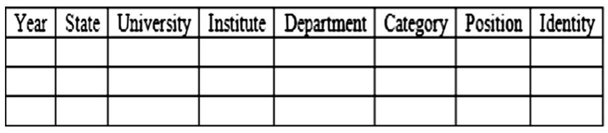
4.1.3. Meta data

A table is created to store Meta data in each DW that built's at any location of any levels in an 8-levels hierarchal structure.

4.2. Construction of various levels in case study

4.2.1. Construction of DW at Level Year

The Level Year is the first level in our case study. A DW named 2010 that built at Level Year at a location Y: 10. The DW built at this level using Algorithm 1. The DW build at this level is numbered according to the method discussed in IDDW



Year	State	University	Institute	Department	Category	Position	Identity

Figure 6 The Common table in each DW built in an 8-level hierarchical structure for case study.

modeling in section three. As only one DW is build at this level it is numbered as D_1^1. At this stage DW named 2010 is the single DW in an 8-level hierarchical structure. The Common table of it is populated with all the set of records of users registering in the system.

4.2.2. Construction of DW at Level State

The Level State has 100 locations available, numbering from $S:00$ to $S:99$, for building DW's. There are 35 states and union territories in India so only 35 DW's need to be build at Level 2, in the locations from $S:01$ to $S:35$. The DW is built at this level using Algorithm 2. The DW build is named after the name of the state that builds it.

One more DW is built at the location $S:00$. This DW stores the information about the people of Country level such as Prime Minister, Minister of Higher Education. The records concerned to this DW previously stored in the Common table of DW, 2010 at Level 1 are passed to it.

A state Uttar Pradesh builds its DW named UP at the location $S:01$. All the records with the value "Uttar Pradesh" in the second field of records in the Common table of DW at $Y:10$ is transferred to the Common table of DW, UP. Once the set of records are transferred the Meta Data of DW, 2010 at $Y:10$ is updated.

4.2.3. Construction of DW at Level University

The Level University has 100 locations available, numbering from $U:00$ to $U:99$, for building DW's. For every location in the Level State such 100 location are available. Each of the different state has different number of universities. Thus there are different numbers of locations occupied in the Level University for each state. Either, Algorithm 2 or Algorithm 3 is used to build any DW at this level. The algorithm to be used depends on whether there is a DW above it in the Level State or not.

The state Uttar Pradesh, already having a DW named UP, has around 40 universities. Any university in Uttar Pradesh, that builds its own DW uses Algorithm 2. The maximum number of DW's that may be built for the universities in Uttar Pradesh is 40 in the locations from $U:01$ to $U:40$.

One more DW is build at the location $U:00$. This DW will store information about the people of that state such as Chief Minister, Minister of Higher Education. The records concerned with this DW stored in the Common table of DW UP are passed to it. A university, of state Uttar Pradesh, named Gautam Budh Technical University builds its own DW named "GBTU" at the location $U:01$. All the records with the value "Uttar Pradesh" in the second field and the value "Gautam Budh Technical University" in the third field of records in the Common table of DW, UP at $S:01$ are transferred to the Common table of DW, GBTU. Once the set of records are transferred the Meta Data of DW, UP is updated.

4.2.4. Construction of DW at Level Institute

The Level Institute has 1000 locations available for each location in the Level University, numbering from $I:000$ to $I:999$, for building DW's. Each of the different university has different number of institutes. Thus there are different numbers of locations occupied in the Level Institute for each university. Either Algorithm 2 or Algorithm 3 is used to build any DW at this level. The algorithm to be used depends on whether there is a DW above it in the Level University or not.

The university Gautam Budh Technical University, already having a DW named "GBTU", has around 700 institutes. Any institute in Gautam Budh Technical University, that builds its own DW uses Algorithm 2. The maximum numbers of DW's that can be built for the institutes in Gautam Budh Technical University is 700 in the locations from $I:001$ to $I:700$.

One more DW is build at the location $I:000$. This DW will store information about the people of that university such as Chancellor, Vice Chancellor, Registrar. The records concerned with this DW stored in the Common table of DW, GBTU are passed to it. An institute, of university Gautam Budh Technical University, named Kamla Nehru Institute of Technology builds its own DW named "KNIT" at the location $I:001$. All the records with the value "Gautam Budh Technical University" in the third field and the value "Kamla Nehru Institute of Technology" in the fourth field of records in the Common table of DW, GBTU at $U:01$ are transferred to the Common table of DW, KNIT. Once the set of records is transferred the Meta Data of DW, GBTU is updated.

4.2.5. Construction of DW at Level Department

The Level Department has 100 locations available for each location in the Level Institute, numbering from $D:00$ to $D:99$, for building DW's. Each of the different institute has different number of departments. Thus there are different numbers of locations occupied in the Level Department for each institute. Either Algorithm 2 or Algorithm 3 is used to build any DW at this level. The algorithm to be used depends on whether there is a DW above it in the Level Institute or not.

The institute Kamla Nehru Institute of Technology, already having a DW named KNIT, has around 20 departments. Any department of Kamla Nehru Institute of Technology, that builds its own DW uses Algorithm 2. The maximum numbers of DW's that can be built for the departments in Kamla Nehru Institute of Technology is 20 in the locations from $D:01$ to $D:20$.

One more DW is build at the location $D:00$. This DW will store information about the people of that institute such as Director, Registrar. The records concerned with this DW stored in the Common table of DW, KNIT are passed to it. A department, of institute Kamla Nehru Institute of Technology, named Computer Science & Engineering builds its own DW named CSE at the location $D:01$. All the records with the value "Kamla Nehru Institute of Technology" in the fourth field and the value "Computer Science & Engineering" in the fifth field of records in the Common table of DW, KNIT at $I:001$ are transferred to the Common table of DW, CSE. Once the set of records are transferred the Meta Data of DW, KNIT is updated.

4.2.6. Construction of DW at Level Category

The Level Category has 10 locations available for each location in the Level Department, numbering from $C:0$ to $C:9$, for building DW's. Each of the different department has different number of categories. Thus there are different numbers of locations occupied in the Level Category for each department. Either Algorithm 2 or Algorithm 3 is used to build any DW at this level. The algorithm to be used depends on whether there is a DW above it in the Level Department or not.

The department Computer Science & Engineering, already having a DW named CSE, has around 5 categories. Any category in Computer Science & Engineering, that builds its own DW uses Algorithm 2. The maximum numbers of DW's that may be built for the categories in Computer Science & Engineering department are 5 in the locations from $C:1$ to $C:5$.

One more DW is build at the location $C:0$. This DW will store information about the people of that department like HOD. The records concerned with this DW stored in the Common table of DW CSE are passed to it. A category, of department Computer Science & Engineering, named Faculty builds its own DW named FAC at the location $C:1$. All the records with the value "Computer Science & Engineering" in the fifth field and the value "Faculty" in the sixth field of records in the Common table of DW, CSE at $D:01$ are transferred to the Common table of DW, FAC. Once the set of records is transferred the Meta Data of DW, CSE is updated.

4.2.7. Construction of DW at Level Position

The Level Position has 10 locations available for each location in the Level Category, numbering from $P:0$ to $P:9$, for building DW's. Each of the different categories has different number of positions. Thus there are different numbers of locations occupied in the Level Position for each category. Either Algorithm 2 or Algorithm 3 is used to build any DW at this level. The algorithm to be used depends on whether there is a DW above it in the Level Category or not.

The category Faculty, already having a DW named FAC, has around 4 positions. Any position in category Faculty, that builds its own DW uses Algorithm 2. The maximum numbers of DW's that can be built for the positions in the Category Faculty is 4 in the locations $P:1$ to $P:4$.

One more DW is build at the location $P:0$. A position, of category Faculty, named Professor builds its DW named PROF at the location $P:1$. All the records with the value "Faculty" in the sixth field and the value "Professor" in the seventh field of records in the Common table of DW, FAC at $C:1$ are transferred to the Common table of DW, PROF. Once the set of records is transferred the Meta Data of DW, FAC is updated.

4.2.8. Construction of Level Identity

The last level, named Identity, does not build any data warehouse at any of its location. Rather, a process is executed at this level to generate a unique three digits number. The number is generated serially at each location in this level. For one location of Level Position, 999 unique numbers (from 001 to 999) are generated. The three digit number acts as a unique Id for a user. The unique Id helps to distinguish any two users in a

Year: 2010 | State: Uttar Pradesh | University: Uttar Pradesh Technical University | Institute: Kamla Nehru Institute of Technology | Department: Computer Science & Engineering | Category: Faculty | Position: Professor | Identity as 002

Figure 7 A hierarchy.

hierarchy even if all the locations of a user are same from Level Year to Level Identity.

Once the various levels are constructed, a hierarchy is formed with DW at a location in each level of hierarchy. The hierarchy formed is shown in Fig. 7.

The location in Level Year is *Y:* 10 with DW named 2010. The location in Level State is *S:* 01 (Uttar Pradesh) with DW named UP. The location in Level University is U*:* 01 (Gautam Budh Technical University) with DW named GBTU. The location in Level Institute is *I:* 001 (Kamla Nehru Institute of Technology) with DW named KNIT. The location in Level Department is *D:* 01 (Computer Science & Engineering) with DW named CSE. The location in Level Category is *C:* 1 (Faculty) with DW named FAC. The location in Level Position is *P:* 1 (Professor) with DW named PROF. The identity that is generated at Level Identity for *ID:* 002 is 002.

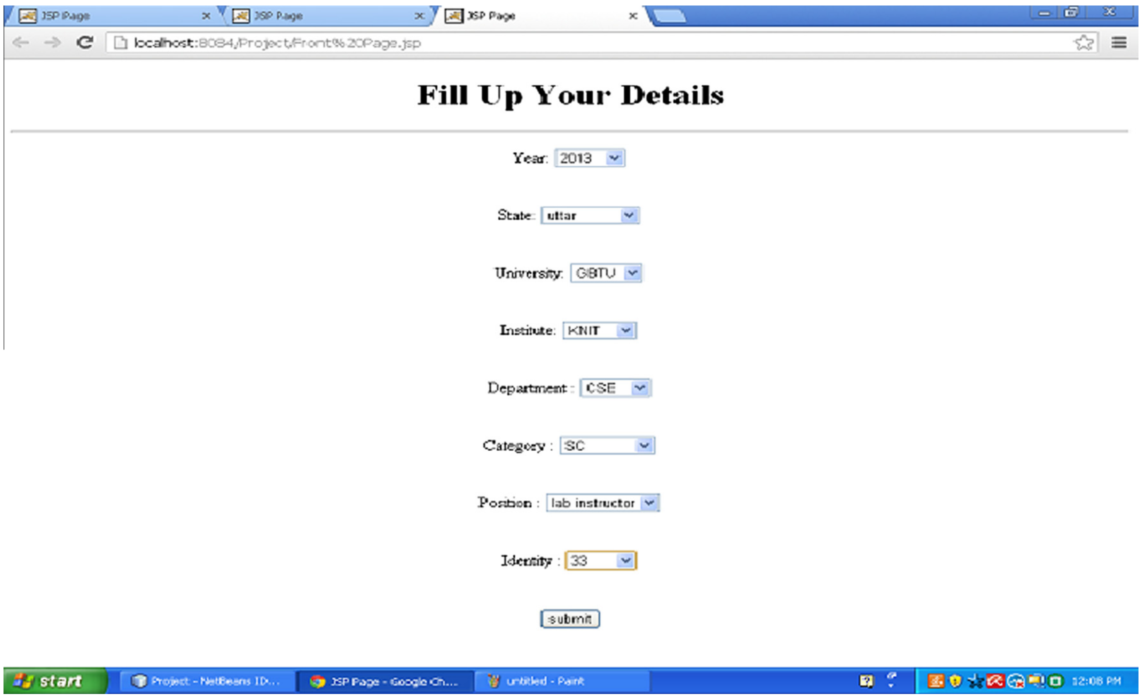
5. Experimental setup

Once the various levels are constructed, a hierarchy is formed with DW at a location in each level of hierarchy. A case study discussed in previous section, shows how an 8-level hierarchical architecture is developed which is analogous to IDDW. To implement the case study, a program is written in JAVA with Net Beans: 6.9 as an IDE, using the Web server: Apache Tomcat 6.0.26. All the tables required in implementation of case study are built in the database MySql 5.0.45. The hardware needed to model the case study, includes a 16 ports Switch and eight machines each with the following specifications: Core 2 Duo processor, 3.0 GHz and 1 GB RAM.

The eight ports of a switch are used to make a hierarchy, of an 8-level, shown in Fig. 4 within blue rectangle. The eight ports represent the eight locations in the hierarchy. The seven different machines may be connected to seven different ports as and when needed. All these seven machines are preloaded with the database, consisting of the data model, discussed previously in case study. The eighth machine is connected to one of another port of the switch. This machine will be used by the users for getting its input. To get user inputs a front page, shown in Fig. 8, is developed.

The front page contains eight drop-down menus. In seven of these, from top, one can also enter new values. In eighth drop-down menu a three digit number is generated automatically. Each drop-down menu is labeled, with name same, as that of each level in 8-level hierarchical architecture considered in case study.

Data for 700 different users are inputted via front page from the specified eighth machine. The data for 700 users entered comprise of 100 users from each of the seven locations in each level of a hierarchy, from top. Initially, out of the seven machines only one is attached to first port of switch representing the location of Level Year. Thus all the values for each of the 700 different users get stored in the Common table of it. The Meta data is also created storing information about the entered records of 700 users. Now, another machine is connected to second port of switch representing the location Uttar Pradesh. At this stage machine attached to first port of switch allocates, 600 records to the machine at second port, via transaction manager. The Meta data of it is created and the Meta data of machine at first port gets updated. The Local MDS (designed and implemented in each of seven machines) of



Fill Up Your Details

Year: 2013

State: uttar

University: GBTU

Institute: KNIT

Department : CSE

Category : SC

Position : lab instructor

Identity : 33

Figure 8 The front page.

machine at first port, transfers the information of allocated records to Global MDS designed and implemented on the machine used for getting user input's.

On the similar basis machines at five different ports are added one after the other. The similar process gets repeated as each machine is added. Once, all seven machines are attached to the ports the Common table of each contains 100 records in it.

A graph shown in Fig. 9 is plotted. This shows the number of records in the Common table of each machine, as, attached to the ports of switch one after the other. The *x*-axis, of graph represents the port number (analogues to a location in a level) to which machine is attached. The *y*-axis represents the number of records. It is deduced from the graph that as the machine (representing DW) is connected to different ports (analogues to a location in a level) the records are distributed from the top most machine to the connected machines. It gives advantage in mining of records from IDDW.

The next section focuses on the various observations that are made about the IDDW. The analysis is also made, in the next section, based on the various observations.

6. Observation & analysis

In this section, we discuss the various observations that are made about the IDDW. The observations are made on the basis of the case study. The observations are also made on the basis of outcomes observed while inputting the user data in the architecture, implemented for the case study. Some of the observations about the system are as follows:

6.1. Ability

The system is able enough for storing and linking millions of records. The system is able so by the Common table present in each DW, that is built, at each location of any level. As there are Other tables also in each DW that is built, it is able to store local data of a location where DW is built. The advantage of it is that, the decision making process at local level becomes fast. The system is also able in getting user inputs in much convenient way through the front page.

6.2. Adaptability

The system is highly adaptable. It is adaptable in the sense that any location defined in it can build its own DW, as and, when needed. It is made possible with the help of internal organization designed for the system.

6.3. Availability

There is high availability of DW at a location, with minimum overhead, from the place of user inputs. The advantage of it is that the process of data getting stored in desired DW becomes fast. The availability increases further as more and more DW's are built at various locations in various levels of IDDW. The same can be seen for the hierarchy implemented for experiment work. In the hierarchy availability of DW increases as more machines, with database installed, are attached to the ports of switch. The percentage of availability of DW is calculated and a graph is plotted shown in Fig. 10.

The value of percentage shows the availability of DW for the case when any one of the DW fails among the DW built so far. It means that DW built, up to, in a location of various levels in a hierarchy starting from top. It can be seen from the graph that percentage availability of DW does not increase much with the increase in DW at more levels in a hierarchy.

6.4. Capability

The system is capable enough to keep track of each and every record about user entered. This is achieved through Global MDS. The Global MDS always keeps the updated status about the DW's where each record gets stored. The Local MDS implemented in each machine provides the updated information to the Global MDS.

6.5. Reliability

The centralized DW has only one DW. The failure of which, fails the entire system. IDDW with records distributed in many DW's the entire system never fails. However, if any DW in it fails only that part of the system fails concerned with this DW. Thus the system is highly reliable as compared to centralized DW. For the analyzing about the reliability the hierarchy shown in Fig. 7 is considered.

We calculated reliability by taking two different cases: worst case and best case: First, Reliability for worst case: In this case we make a machine with DW fails, having maximum number of records in it; Second, Reliability for best case: In this case we make a machine with DW fails, having minimum number of records in it.

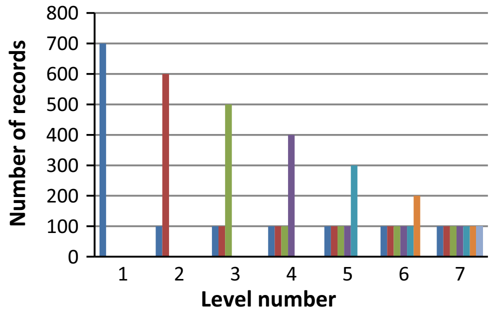


Figure 9 Number of records in the Common table of each machine.

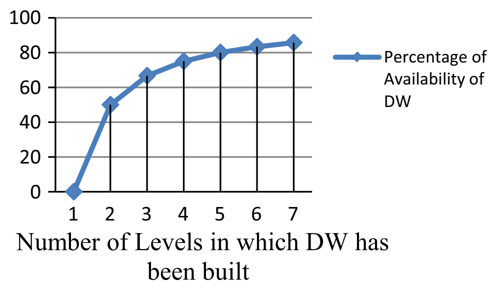


Figure 10 Percentage of Availability of DW.

Table 2 The value of reliability calculated for the two cases.

Number of levels in which DW has been built	Reliability for worst case	Reliability for best case
1	0	0
2	0.142857	0.857143
3	0.285714	0.857143
4	0.428571	0.857143
5	0.571429	0.857143
6	0.714286	0.857143
7	0.857143	0.857143

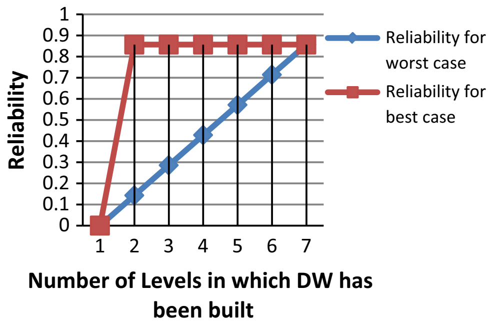


Figure 11 Reliability for best case and worst case.

The different values for reliability are obtained in both by making the DW, fail, out of available DW in a hierarchy.

Table 2 shows the different values of reliability calculated for the two cases. The values for two cases are obtained for the number of levels in a hierarchy, where DW has been built. The graph is plotted for the values given in Table 2.

It can be observed from the graph shown in Fig. 11, that reliability in worst case is improved, as more number of DW built at more number of levels.

6.6. Scalability

The system is highly scalable both vertically and horizontally. Scalability in vertical direction means one can scale-up the system to any number of levels. The DW can easily be built at any level, using Algorithms 2 or 3. Scalability in horizontal direction means one can scale up the system by adding more number of locations in any level. The DW at any new location, increased, can easily be build by using Algorithms 2 or 3. The same can be seen from Table 1 and through a graph shown in Fig. 5, plotted for data in Table 1.

7. Conclusion

In this paper, we proposed a new method for the hierarchical distribution of DW and named it as IDDW. IDDW is mathematically modeled, that helps to number each DW built in IDDW. The IDDW architecture, using and justifying mathematical modeling, is also presented. The data model presented consists of the tables that satisfy the requirement of storing the various data in each DW. A proper internal organization is presented that helps in proper working of IDDW architecture. It also make possible of transferring records from one DW to another. It has been seen experimentally that by using the algorithms presented in the paper a DW is built with ease at any location of any level of IDDW. Finally the proposed architecture is justified by taking a case study. An experimental setup was prepared for case study. The experiment was performed and various observations were made on the basis of result obtained from experiment. The observation reflects about the ability, adaptability, availability, capability, reliability and scalability of IDDW.

8. Future scope

After the implementation of the proposed work, IDDW, there is a scope for the researchers to do brain storming on it. Some of the future scopes on which work may be extended further are as follows: One can find the method of intelligently placing of data in a DW within IDDW. Intelligent placing means locating the correct DW, within IDDW, for the concerned data; One can find the solution of intelligently mining of data from a DW in IDDW; and Parallelization approach may be implemented for both placing of data in IDDW and in mining of data from IDDW.

References

[1] Angela Bonifati, Fabiano Cattaneo, Stefano Ceri, Alfonso Fuggetta, Stefano Paraboschi. Designing Data Marts for Data Warehouses. ACM Trans Softw Eng Methodol 2001;10(4):452–83.

[2] Bernardino J, Madeira H. Experimental evaluation of a new distributed partitioning technique for data warehouses. Int Database Eng Appl Symp 2001:312–21.

[3] Ceri, Pelagatti. Distributed databases: principles and systems. McGraw-Hill; 1984.

[4] Chaki N, Sarkar BB. Virtual data warehouse model- ling using petri nets for distributed decision making. J Conver Inform Technol 2010;5(5):8–21.

[5] Costa M, Madeira H. Handling big dimensions in distributed data warehouses using the DWS technique. DOLAP'04 Washington, DC, USA, November 12–13; 2004. p. 31–37.

[6] Costa R, Furtado P. Data Warehouses in Grids with High QoS. DaWaK, volume 4081 of Lecture Notes in Computer Science. Springer; 2006, pp. 207-217.

[7] Hsiang-Jui Kung, LeeAnn Kung, Adrian Gardiner.Comparing Topdown with Bottom-up Approaches. In: 2012 Proceedings of the Information Systems Educators Conference ISSN: 2167–1435, New Orleans Louisiana, USA v29 n1910.

[8] Ileana ŞTEFAN, Maricel POPA. Distributed Database Design – Top-Down Design, vol. 48, Number 1; 2007.

[9] Inmon WH. Building the data warehouse. John Wiley & Sons; 1993.

[10] Inmon WH. Building the data warehouse (second edition). John Wiley and Sons; 1996.

[11] Jarke M, Lenzerini M, Vaasssiliou Y, Vassiliadis P. Fundamentals of data warehouse 2000.

[12] Karima T, Abdelatif A, Habib O. Data warehouse decentralization strategy. IEEE Int Confer E- business Eng 2010:72–6.

[13] Lawrence M, Rau-Chaplin A. The OLAP-enabled grid: model and query processing algorithms. In: Proceedings of the 20th international symposium on high performance computing systems and applications (HPCS'06), Canada; May 2006.

[14] Noaman AY, Barker K. Distributed data warehouse architectures. J Data Warehousing 1997;2(2):37–50. April.

[15] Noaman AY, Barker K. A horizontal fragmentation algorithm for fact relation in a distributed data warehouse. In: Proceedings

of the eight international conference on information and knowledge management (CIKM'99); November 1999. p. 154–161.

[16] Ozsu, Valduriez. Principles of distributed database systems. Prantice Hall; 1991.

[17] Bhati Ruby, et al. Distributed database system: the current features and problems. In: International Journal of Computer Science and Management Research, vol. 2 Issue 3; March 2013. ISSN 2278–733X.

[18] Shailesh R, Thakare CA, Dhawale, Gadicha Ajay B. Design distributed database strategies for SQMD architecture. (IJEAT) ISSN: 2249–8958, vol. 1, Issue-2; 2011.

[19] Su H, Tang C, Qiao S, Li C, Zhang T, Dai S. Construct distributed RFID data warehouse based on concept hierarchy. In: IEEE Internnational Workshop on Anti-Counterfeiting, Security, Identification, Xiamen, Fujian; 2007. p. 461–463.

[20] Wehrle P, Miquel M, Tchounikine A. A model for distributing and querying a data warehouse on a computing grid. In: 11th International conference on parallel and distributed systems (ICPADS'05), vol. 1; 2005. p. 203–209.

[21] Wehrle P, Miquel M, Tchounikine A. A grid services-oriented architecture for efficient operation of distributed data warehouses on globus. In: 21st International Conference on Advanced Networking and Applications (AINA '07) Canada; May 2007. p. 994–999.

[22] White C. A technical architecture for data warehousing. InfoDB J 1995;9(1):5–11.

[23] Zhao J, Schewe K-D. Using abstract state machines for distributed data warehouse design. In: Hartmann S, Roddick J, editors. First Asia-Pacific Conference on Conceptual Modelling (APCCM 2004), vol. 31 of CRPIT, Australian Computer Society, Dunedin, New Zealand; 2004. p. 49–58.

[24] Zhou S, Zhou A, Tao X, Hu Y. Hierarchically distributed data warehouse. IEEE J 2000:848–53.

Agarwal Abhay Kumar is a Assistant Professor in Computer Science & Engineering Department at Kamla Nehru Institute of Technology (KNIT), Sultanpur, India. He received his B.Tech degree in 1999 from (BIET) Jhansi in Electronics & Instrumentation Engineering and M. Tech in 2006 from Samrat Ashok Technology Institute (SATI), Vidisha, in Information Technology. Presently he is pursuing his Ph.D. in Computer Science & Engineering. His research interests are Data warehouse & Data Mining, Parallel Processing and Algorithms.

Badal Neelendra is a Assistant Professor in Computer Science & Engineering Department at Kamla Nehru Institute of Technology (KNIT), Sultanpur, India. He received his B.Tech degree in 1997 from (BIET) Jhansi in Computer Science & Engineering, M.E. (2001) in Communication, Control and Networking from Madhav Institute of Technology and Science (MITS), Gwalior, and Ph.D. (2009) in Computer Science & Engineering from Motilal Nehru National Institute of Technology (MNNIT), Allahabad. He is Chartered Engineer (CE) from Institute of Engineers (IE), India. He is Life Member of IE, IETE, ISTE and CSI-India. He has published about 30 papers in International/ National Journals, conferences and seminars. His research interests are Distributed System, Parallel Processing, GIS, Data Warehouse & Data Mining, Software Engineering and Networking.