Optimization of ETL Work Flow in Data Warehouse

Kommineni Sivaganesh
M.Tech Student ,
CSE Department ,
Anil Neerukonda Institute of Technology & Science
Visakhapatnam, India.
Sivaganesh07@gmail.com

P Srinivasu
Associate Professor,
CSE Department,
Anil Neerukonda Institute of Technology & Science
Visakhapatnam, India.

Dr Suresh Chandra Satapathy
Professor and H.O.D
CSE Department,
Anil Neerukonda Institute of Technology & Science
Visakhapatnam, India.

***Abstract*— ETL is responsible for extracting the information or data from different several Areas and applies some cleaning, customization, transformation function for data and finally loading into the data warehouse. This paper presents , to implement the one ETL scenario with the help of ARKTOS II. It is a stepwise process, firstly to design the conceptual model for the ETL scenario it contains the quick documentation which contains their relationships, attributes and transformation among them. Next one is a formal logical model for the ETL scenario, it contains the flow of data from the sources towards the data warehouse through the transformations and data stores. ETL processes handle the large volume of data, and managing the workload. It is a complex task and expensive operations in terms of time and system resources. Therefore to minimize the time required for completion of ETL workflow and resources needed for the ETL tasks. So that we implement the optimization for the ETL workflows, in the minimizing the total cost of the ETL scenario.**

Keywords- Extraction, Transformation, Loading, Conceptual Design, Logical Design, Optimization.

I. Introduction

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It contains previous historical data derived from multiple heterogeneous sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources. According toInmon [1], a DW is “a collection of subject-oriented, integrated, non-volatile and time variant data in support of management decisions”.

The major characteristic of the data warehouse is

- Subject Oriented
- Integrated
- Nonvolatile
- Time Variant

A Subject oriented

Data warehouse concentrates on major subjects for decision making, the subjects are customer, product, sales, supplier rather than concentrating on day to day transactions. Data warehouse mainly focusing on the modeling and analysis of the data [1].

B Integrated

A data warehouse is constructed by the integrating the multiple heterogeneous sources such as flat files, relational databases. Before loading into the data warehouse , preprocessing techniques can apply to that data, that are data cleaning and data integration , in data cleaning process to remove the irrelevant and noisy from the data, here binning, regression and clustering techniques applies . In the data integration process there are no of issues raised that are schema integration and object matching i.e. How the real world entities can be matched up. This can be solved by using the metadata. Next one is redundancy, the repeated storage of data; it can solve by using the correlation analysis [1].

C Nonvolatile:

Non-volatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable you to analyze what has occurred [1].

D Time variant

Data warehouse main key element is time variant , it is very important, in this data are stored in or to access the data from the data warehouse the past years, its measurement is time, it is useful in business, the trend changes [1].

Data warehouse architecture:

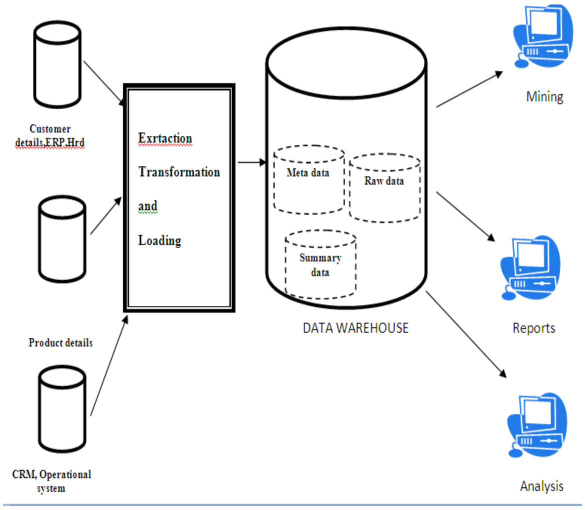


Fig 1: Data warehouse architecture

II. ETL

To manage the data warehouse operations, specialized tools are available in the market, called ETL tools. [3] *ETL (Extraction-Transformation-Loading) tools* are a category of software tools responsible for the extraction of data from distributed sources, their cleansing [4] and customization and finally their loading into the data warehouse. [8] [9] [10] [11]

First of all to know how ETL processes set into the data warehouse. We have Three stages in the ETL processes.

First Stage: In this process, we have to collect the requirements from the users , understand the problem specification. Normally this stage is called the conceptual design , it is used to collect the data , attributes , relationships and transformation between them.

Second Stage: In this process, we design the logical model of the ETL workflows. It concentrates on the data is flowing from different sources and finally loaded into the data warehouse through some activities.

Third Stage: In this process , to optimize the designed ETL workflows , there are different optimization techniques can be used to optimize the workflow, the best is heuristic search.

III . Conceptual Design

In this we present the conceptual model for the ETL workflows [2] [5] [6]. It is the first stage of the ETL process .It gives the brief idea about the problem statement. Here we collect the data from the different sources and model that data finally loaded into the data warehouse.

We focus on the conceptual part of the definition of the ETL processes. More specifically, we are dealing with the early stages of the data warehouse design. During this period, the data warehouse designer has considered with two tasks which are practically executed in parallel:

(a) The collection of requirements for the part of the users.

(b) The analysis of the structure and content of the existing data sources and their intentional mapping to the common data warehouse model. The design of an ETL process aims at the production of a crucial deliverable:

the mapping of the attributes of the data sources to the attributes of the data warehouse tables.

EXAMPLE:

To motivate our discussion , we introduce one example. The scenario involves the propagation of data from the concept PARTS of source S1 as well as from the concept PARTS of source S2 to the data warehouse. In the data warehouse, DW.PARTS stores (PRODUCT_ID, SALE_ITEM, CUSTOMER, DATE, SALE_YEARS_COST). We assume that the first supplier is India and the second is American, thus the data coming from the second source need to be converted the sales of month into sales of years by aggregation function, similarly in s1 also. In India the cost can calculate in terms of rupees, but in America calculated in terms of dollars, so that we use the transformation functions. We have to convert the cost from us dollars to Indian rupees. The daily sales can be identified with the unique date format. America has one date format, and the India is different, so that we change the date formats. Our example source s1 having an India date format and source s2 having in America date format, now we have changed the American date format in India date format.

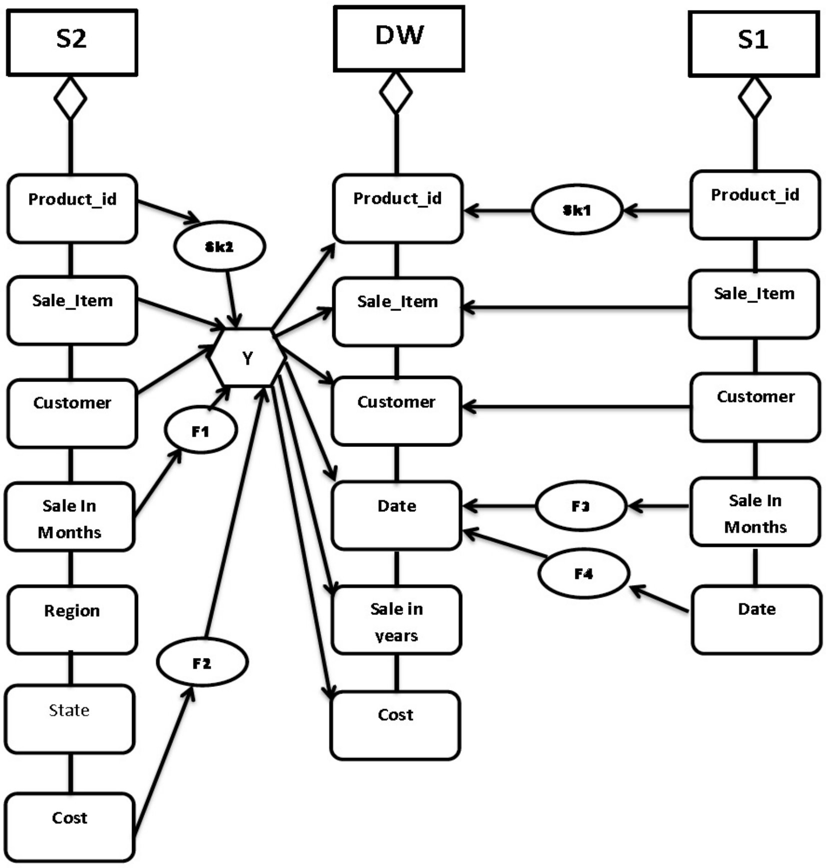
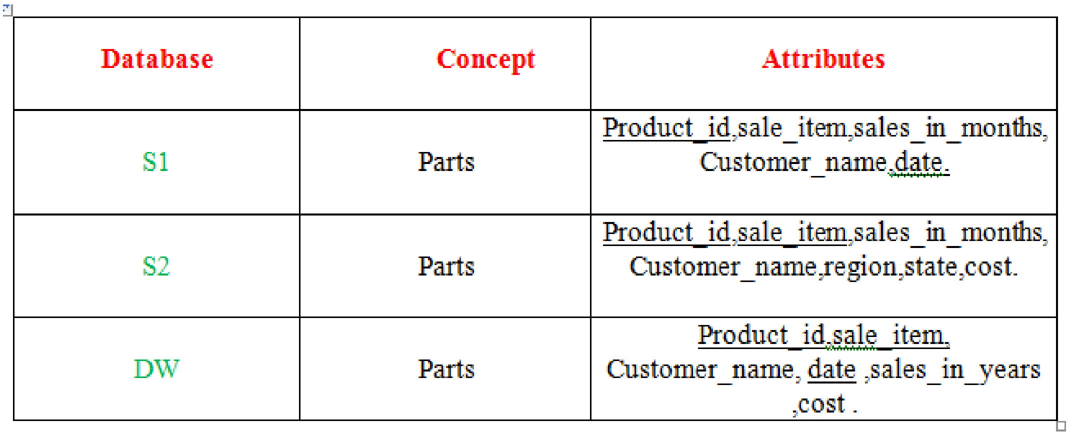


Fig1: Conceptual design of our example.

Table 1 : The schemata of the source databases and of the data warehouse.



Database	Concept	Attributes
S1	Parts	Product_id,sale_item,sales_in_months, Customer_name,date.
S2	Parts	Product_id,sale_item,sales_in_months, Customer_name,region,state,cost.
DW	Parts	Product_id,sale_item, Customer_name, date ,sales_in_years ,cost .

IV. Logical Model

To achieve the data centric work flows, we present the logical Meta model for abstraction of the ETL work flows. The data stores, activities and their constituent parts are formally defined. The main aim of the Logical model for activities of ETL work flows, its concentration on the flow of data from different sources towards Data Warehouse by using Data stores, activities and set of relationships. An activity is an event with input schema, output schema and provider schema[2] [5] [6].

The overall implementation of ETL scenario involving Activities, Data sets and functions are deployed in a graph. This graph is known's as 'Architectural graph'. This graph is implemented by using the ARKTOS-II tool. It provides some rules which must be obeyed. All entities are represented as nodes and relationships represented as the edges. The ARKTOS-II tool provides basic entities for exiting model:

***Attributes and part of relationships*:**

The first step of the architectural graph is structured entities along with attributes. Later incorporation with the function along with part of relationship.

Data types and instance-of relationships :

Which type the entity correspond to.

Parameters and regulator relationships :

To establish the regulator relationships of the scenario , we provide the link the parameters of activities to the terms i.e. Attributes or constants.

***Provider relationships*:**

The final step is to add the provider relationships to the Architectural graph it captures the flow of data from sources towards the record sets in the data warehouse.

Example:

The scenario involves the propagation of data from the concept PARTS of source S1 as well as from the concept PARTS of source S2 to the data warehouse. In the data warehouse, DW.PARTS stores (PRODUCT_ID, SALE_ITEM, CUSTOMER, DATE, SALE_YEARS_COST).

We assume that the first supplier is India and the second is American, thus the data coming from the second source need to be converted the sales of month into sales of years by aggregation function, similarly in s1 also. We have to convert the cost from us dollars to Indian rupees.

Where as in S1 we have to convert the date format into an Indian date format and S1 we have to convert the sales of the month into sale of years.

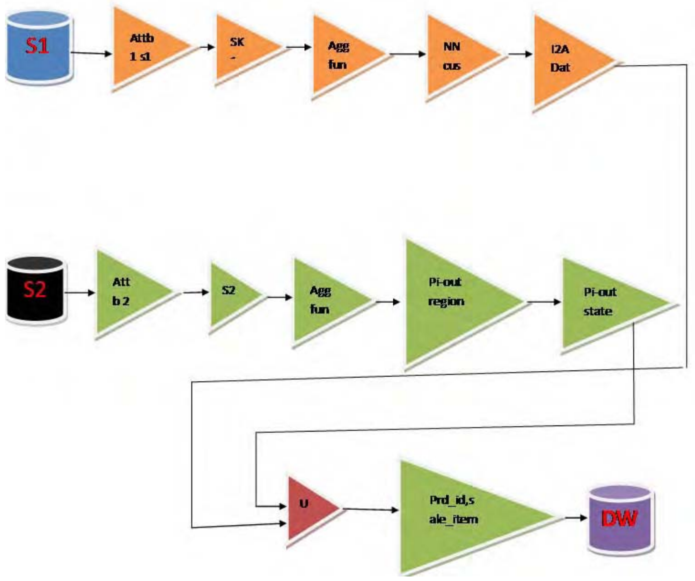


Fig 2: logical design of our example.

V. Optimization

ETL scenario must be completed within a certain time window, but it takes a lot of time to execute it, so that in order to improve the efficiency of the ETL works flow [3]. In this section we present an optimization of ETL processes. First of all identify the problem and next define the statement of our problem as a state search problem, and finally discuss the some guidelines to correct formulation of the states before and after execution. There are three algorithms are implemented for the optimization of the ETL workflows. Initially we construct the search space where ever it possible, and to reduce the search space by using Heuristic algorithm, finally produce the optimistic results.In this section , we present the theoretical query optimization for the problem. This problem can be modelled as a state search problem , represented as a graph . It contains set of nodes and edges. The nodes of the graph represent activities and record stores , edges can be represent the flow of data among the nodes. Next transitions applied to the traditional optimization for extending. Finally providing the algorithms towards optimization of ETL workflows initial exhaustive algorithm to explore the state search problem, to reduce the search space we introduce the greedy and heuristic algorithms in order to improve the efficiency of the workflow.

Formal statement of the problem:

In this ETL workflow optimization can be modeled as a state space search problem. First, we give a formal definition of the ETL workflow, in this an ETL workflow can be represented as a DAG (directed acyclic graph) consist of a set of nodes and edges. The nodes can be denoted as record sets and activities, here record is nothing but data storage i.e. Flat files. An edge can be denoted as a data provider. In our motivation we will model the ETL workflow as a graph, so that we assume the list of activities A , set of record sets R provider relationships p .

Formally ETL workflow can be viewed as the DAG, actually the graph contains.

$$G\ (V, E).$$
$$V= A \text{ u } R\ .\quad \text{u= union operator.}\quad E= p\ .$$

Next we present the problem of the ETL workflow , in this section states and transitions described , each state is described as a graph. Transitions T is used to generate new, equivalent states. Here equivalent states are the states that based on the same input produce the same output, this can be achieved by the following ways.

1. By replacing common tasks.
2. By interchanging two activities of the workflow in terms of their execution sequence.
3. By dividing the tasks of workflow.

After that applying the set of logical transitions. Therefore S'=T (S). I.e transition from state S to state S' and describe the states. Then, we define a set of transitions that can be applied to the states in order to produce the search space. Finally, we formulate the problem of the optimization of an ETL workflow.

Searching algorithms:

In this section we present three algorithms towards optimization of ETL workflow.

1. Exhaustive algorithm
2. Heuristic algorithm
3. Greedy-Heuristic algorithm

Exhaustive algorithm:

This is also known as brute-force search. It is simple to implement. In this approach generate the all possible states that can be generated by the all possible transitions apply to the states. Basically a state space can be modelled as a graph, nodes are the states and edges are the transitions from one state to another. The *Exhaustive Search* algorithm (ES) employs a set of *unvisited* nodes, which remain to be explored and a set of *visits* nodes that have already been explored. While there are still nodes to being explored, the algorithm picks an unvisited state and produces its children to be checked in the sequel. The search space is obviously finite and it is straightforward that the algorithm generates all possible states and then terminates. Afterwards, we search the visited states and we choose the one with the minimal cost as the solution to our problem [7].

Heuristic algorithm:

A heuristic is a rule of thumb that leads to a solution. Actually it finds a good solution in reasonable time but that solution is might not be the best. In this to avoid exploring of full state space, we employ set of heuristics based on simple observations and previous experience [7].

Heuristic 1: Factorization (FAC)

This heuristic can be applied to the new state instead of the old state. Need not required the factorization for all the acivitites . The factorization is the process of homologous of two activities a1 and a2 replaced by a new activity a does the same job . It follows the binary operation.

Heuristic 2: Distributive (DIS)

This heuristic clearly says that the new activity is generated from the old activity through DIS operation. In this DIS binary operation involves generating the new activity a1 and a2 old state a.

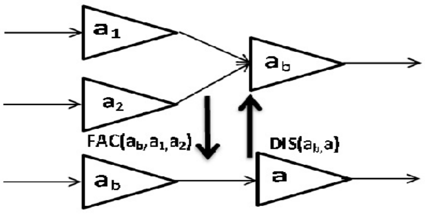


Fig 3 (a): Factorize and Distribute

Heuristic 3: Swapping of the activities.

This heuristic tells that the interchanging of the two activities in the flow. This is unary operation . This operation actually implemented after examining the process of identifying the occurrences of the swap concepts in between the two activities. There are two activities let us say a1 and a2 , next identifying the occurrences of swap, if exists in between them swap (a1, a2) i.e. Execution process is started with a2 and then a1. This is equivalent to actual workflow . It does not affect to the workflow. This operation is to optimize the actual ETL work flow i.e. To reduce the execution time [7].

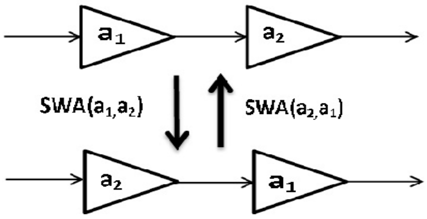


Fig 3 (b): SWAP.

Heuristic 4: Split and Merging

This heuristic reduces the search space. Here merging means combine the two activities a1 and a2 into new activity a_{1+2} [7]. The splitting of the activity a into two activities a1 and a2.

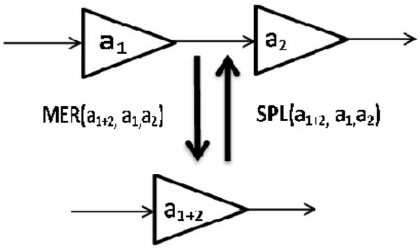


Fig 3 (c): Merge and Split.

Heuristic 5: Divide and Conquer

In this heuristic optimization techniques can apply to part of itself instead of the whole graph. In this process the state can be divided into local groups. The local group is a subset of the graph, which form a linear path of unary activities with elements [7].

The input of the heuristic is initial state s_0

Preprocessing stage: in this phase first all the activities of the ETL workflow constrains are merged. The constrains semantics of the activities and user defined activities, and HS finds the homologous activities (H) and local groups (L) for each state.

Phase 1: In this phase heuristic search proceeds with the SWAP transitions of each local group. In this to find out the better cost of the state than the already minimum cost existed already. It assigns to the S_{min} . The final output of this phase is stated with a global minimum cost [7].

Phase 2: In this phase Heuristic search checks the possible communications between two adjacent local groups . Then apply the factors binary function. HS adds every new state to the visited list and keeps a record of a state with a new minimum cost [7].

Phase 3 : Here the HS finds the new state of phase 2 for activities that could be distributed with binary function. But there is a problem because of the activities of the phase 2 already vectorizing, that activities are not eligible for the again distributing [7].

Phase 4: In this phase HS repeats the process from phase 1 to find the all possible states by using the swap transitions for all the nodes having the local group for updating the S_{MIN} . After completion of previous phases now HS split transition in order to split the all the activities that were already merged.

Finally HS return the state with minimum cost. [7]

Greedy-Heuristic search algorithm:

In this algorithm to increase the performance of heuristic search. Actually in heuristic search the phase1 is repetition in phase4. To improve the performance HS swaps only those that lead to a state with less cost that the existing minimum instead of all pairs of activities for each local group. Then only HS becomes greedy [7].

VI . Experimental results

In this section we have to validate the experimented results. We implement the search algorithms in c++ language, Experimented on the variation of measures Like time, volume of visited states, and improvement of the solution and the quality of the proposed workflow. We have used the simple cost model . There are different ETL workflows can be categorized as small, medium and large ranges. After we check all the test cases for three algorithms will be given best optimum solution according to their visited states and execution time. ES algorithm is slower than the others, it takes a lot of time to execute it. It is suitable for the small ranges of test cases of the ETL workflow gives the optimum solution. Heuristic and Greedy Heuristic both algorithms give the best optimum solution according their quality and execution time . But in practice the best optimum solution is given by heuristic search only. Greedy also gives the same optimal solution as slight changes in it. But the execution time is very fast in Greedy HS compared to the HS.

Algorithm	Exhaustive	Heuristic
Initial Cost	36895	36895
Minimum Cost	32745	32745
States Processed	42	20
States Visited	42	8
Completion Time (sec)	1.95	0.16

VII. Conclusion

In this paper we have to develop the our own ETL scenario . First we implement the framework for the taking the data from the different sources. Next design the conceptual model for the data just like as a blue print of the ETL work flows. After that we design the logical model for this , and then mapping from conceptual model to the logical model. We optimize the scenario here taking the problem can be modeled as a state search problem represented as a graph. We define the some transitions from one state to another state, discuss some issues for state generation. Finally we present the some algorithms to increase the performance the system and which is the best algorithm for chosen based on the experimental results provides in the list.

References

[1] W. Inmon, Building the Data Warehouse, John Wiley & Sons, Inc. 2002.

[2] Alkis Simitsis, Nat. Tech. Univ. Of Athens, Panos VassiliadisUniversity of Ioannina. Optimizing ETL Processes in Data Warehouses.

[3] P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terroviti A Framework for the Design of ETL Scenarios. CAiSE'03, Klagenfurt, Austria, 2k3

[4] E. Rahm, H. Do. Data Cleaning: Problems and Current Approaches. Bulletin of the Technical Committee on Data Engineering, 23 (4),2000.

[5] N. Tryfona, F. Busborg, J.G.B. Christiansen. Starr: A Conceptual Model for Data Warehouse Design. In Proceedings of 2nd ACM International Workshop on Data Warehousing and OLAP (DOLAP '99), Kansas City, Missouri, USA, 1999.

[6] J. Trujillo, S. Luján-Mora. A UML Based Approach for Modeling ETL Processes in Data Warehouses. In the Proceedings of the 22nd International Conference on Conceptual Modeling (ER'03), LNCS 2813, pp. 307–320, Chicago, Illinois, USA.

[7] Alkis Simitsis Panos Vassiliadis Timos Sellis . Optimizing ETL Processes in Data Warehouses.

[8] IBM. IBM Data Warehouse Manager. Available at www-3.ibm.com/software/data/db2/datawarehouse.

[9] Informatica. PowerCenter. Available at: www.informatica.com/products/data+integration/powercenter/default.htm.

[10] Microsoft. Data Transformation Services. Available at www.microsoft.com.

[11] Oracle Corp. Oracle9i™ Warehouse Builder User's Guide, Release 9.0.2. November 2001. Available at: http://otn.oracle.com/products/warehouse/content.html.