Enterprise Planning Process Based On Etl Technology Using Mesh Algorithm for Magna Energy and Hydro Systems

Mrs.S.SUDHA[1], V.S.Saranya[2]
1. Asst.Professor, 2.Final MCA.Department of Computer Applications
Adhiparasakthi Engineering College, Melmaruvathur.

Abstract
Active data warehouses have emerged as a new business intelligence paradigm where data is refreshed in real-time. The changes required in the implementation of Extract Transform Load (ETL) operations which now need to be executed in an online fashion. ETL Transformations includes the join between incoming stream of updates and a disk resident table of historical data. In this context, a Mesh Join algorithm can be mainly proposed for two objectives: i) Maximizing throughput under a specific memory budget and ii) Minimizing memory consumption. The existing ETL transformation uses Semi-Stream Index Join (SSIJ) algorithm for implementing join operations. This algorithm can be very useful for dynamically changing memory space and it contain two objectives such as increasing the streaming arrival rate and data distribution. The problem in the SSIJ algorithm is, it doesn't work well in the arrival of even faster stream of input. The solution for the problem is, to use the Mesh Join algorithm for the implementation of ETL transformation.

Introduction

Managing an organization requires to monitor and log the processed information. These processes will help them to improve their business, customer relationship and more time will be reduced compare to their earlier processes. Generally Enterprises Resources Planning (ERP) tool will help an organization to improve their quality of business. The *ERP* contains six modules; each module has some sub modules also. Modules are related to *User & Customer management, Sales & Services Management, mailing solution and reports*. The highlighted feature of ETL system is the Business Intelligence which used for efficient decision making.

Generally nature of ETL process is receives valid information from different sources, then organize those data in some common format for decision making using any of the ETL Process algorithm

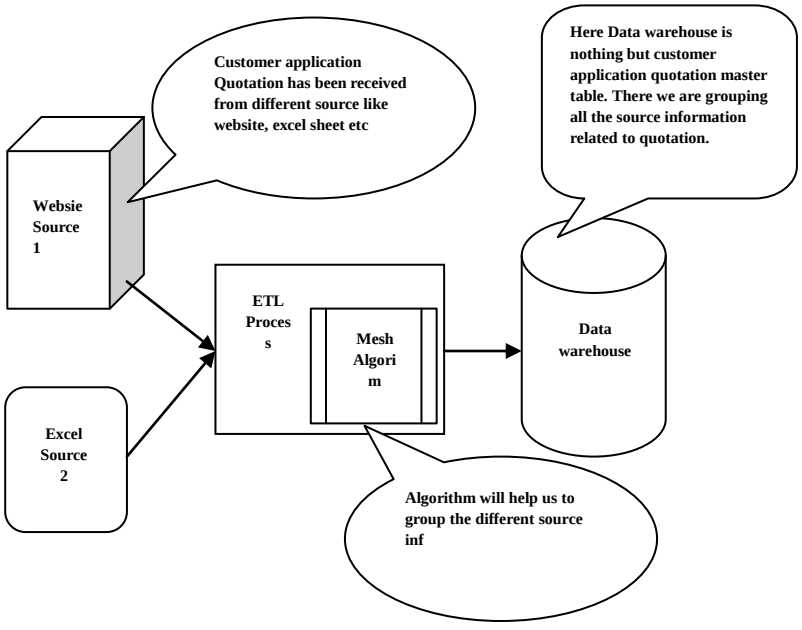

Fig. a) ETL process

Here ETL Process implementation will be done by using **Mesh Algorithm** to overcome *transfer rate* and *memory related* issues.

Work Process

A specialized join algorithm used for ETL transformation is, termed MESH JOIN, that joins a fast update stream *S* with a large disk resident relation *R* under the assumption of limited memory. This is a core problem for active ETL transformations and its solution is thus an important step toward realizing the vision of active data warehouses. MESHJOIN applies to a broad range of practical configurations: it makes no assumption of any order in either the stream or the relation; no indexes are necessarily present; the algorithm uses limited memory to allow multiple operations to operate simultaneously; the join condition is arbitrary (equality, similarity, range, etc.); the join relationship is general and the result is exact.

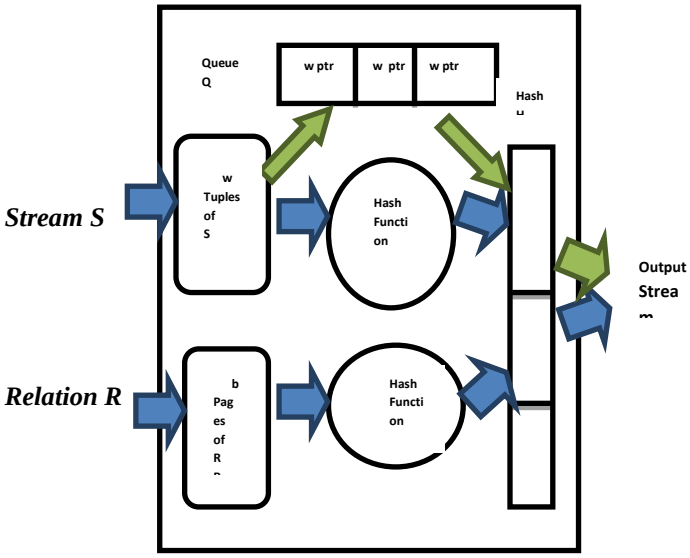

Fig b. Overview of Mesh Join

Mesh Join performs the continuous scan of*R* with an input buffer of *b* pages Stream *S*, on the other hand, is accessed in batches of *w* tuples that are inserted in the contents of the sliding window. To efficiently find the matching stream tuples for each *R*-tuple, the algorithm synchronously maintains a hash table *H* for the in-memory *S*-tuples based on their join-key. Finally, queue*Q* contains pointers to the tuples in *H* and essentially records the arrival order of the batches in the current window. This information is used in order to remove the oldest *w* tuples from *H* when they are expired from the window.

Algorithm MESHJOIN
Input: A relation *R* and a stream *S*.
Output: Stream *R join S*.
Parameters: *w* tuples of *S* and *b* pages of *R*.
Method:

```
While (true)
    Read b pages from R and w tuples from S
    If queue Q is full Then
       Dequeue T from Q
       Remove the tuples of hash H that correspond
       to T
    EndIf

   Add the w tuples of S in H
    Enqueue in Q, w pointers to the tuples in H
    For each tuple r in the b pages of R
    Ouput r  join H
EndWhile
```

On each iteration, the algorithm reads *w* newly arrived stream tuples and *b* disk pages of *R*, joins the *R* tuples with the contents of the sliding window, and appends any results to the output buffer.

Conclusion

In this paper, an operation that is commonly encountered in the context of active data warehousing: the join between a fast stream of source updates *S* and a disk-based relation *R* under the constraint of limited memory.

References:

[1] Mihaela A. Bornea, Antonios Deligiannakis,Yannis Kotidis ,Vasilis Vassalos Athens U of Econ and Business, Technical University of Crete, "Semi-Streamed Index Join For Near-Real Time Execution of ETL Transformation".2011.

[2] Shao Dong Chen, Hong Shen, Rodney Topor, "Efficient Parallel Permutation-Based Range- Join Algorithms on Mesh-Connected Computers".

[3] Neoklis Polyzotis Univ. of California – Santa Cruz alkis@cs.ucsc.edu Spiros Skiadopoulos Univ. of Peloponnese spiros@uop.gr Panos Vassiliadis Univ. of Ioannina pvassil@cs.uoi.gr,"Supporting Streaming Updates in an Active Data Warehouse".