Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 4, April-2017

Parallel Data Extraction from Multiple Data Source in Extraction-Transformation-Loading (ETL) Process

Mr. JANI HARSH M. [1]
M.Tech CE
Department of Computer Engineering
C U Shah College of Engineering & Technology,
Gujarat, India.

Asst. Pro. Ms. VIRAL A PAREKH [2]
Department of Computer Engineering
C U Shah College of Engineering & Technology,
Gujarat, India.

ABSTRACT

Information is very important in all the fields. This information is collected from Data Warehouse. Extraction-Transformation-Loading (ETL) Process is used to fill Data Warehouse. In Data Warehouse, data is extracted from multiple data sources. Now a day's data is extracted from one data source at that time other data sources is ideal. After completion of extraction from one data source, second data source will be introduced for data extraction. This process is very time consuming. This paper proposes parallel data extraction method to optimize ETL. The proposed method is an all-in-one solution that supports processing different types of data from operational systems. This method improves ETL flexibility and increase the speed of extraction to the data warehouse. This paper evaluates the proposed method empirically, which shows that it is more efficient and less intrusive than the standard ETL method.

KEYWORD

Data Warehouse, ETL Process, Parallel Data Extraction, Multiple Data Source.

1. Introduction

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process. [3] Data warehouses are central repositories of integrated current and historical data from one or more different disparate sources. It mainly contains historical data derived from transaction data and the current data from other sources. [3] The heart of DWs is the Extraction-Transformation-Loading (ETL) process. [3] Data Warehouse is filled by ETL process. ETL stands for Extraction Transformation & Loading. To facilitate business analysis data from one or more operational systems needs to be extracted and copied into the warehouse. The process of extracting data from source systems and bringing it into the data warehouse is commonly called ETL. [1] As the name suggests, it performs the following three operations −

- Extracts the data from your transactional system which can be an Oracle, Microsoft, or any other relational database,
- Transforms the data by performing data cleansing operations, and then
- Loads the data into the OLAP data Warehouse.

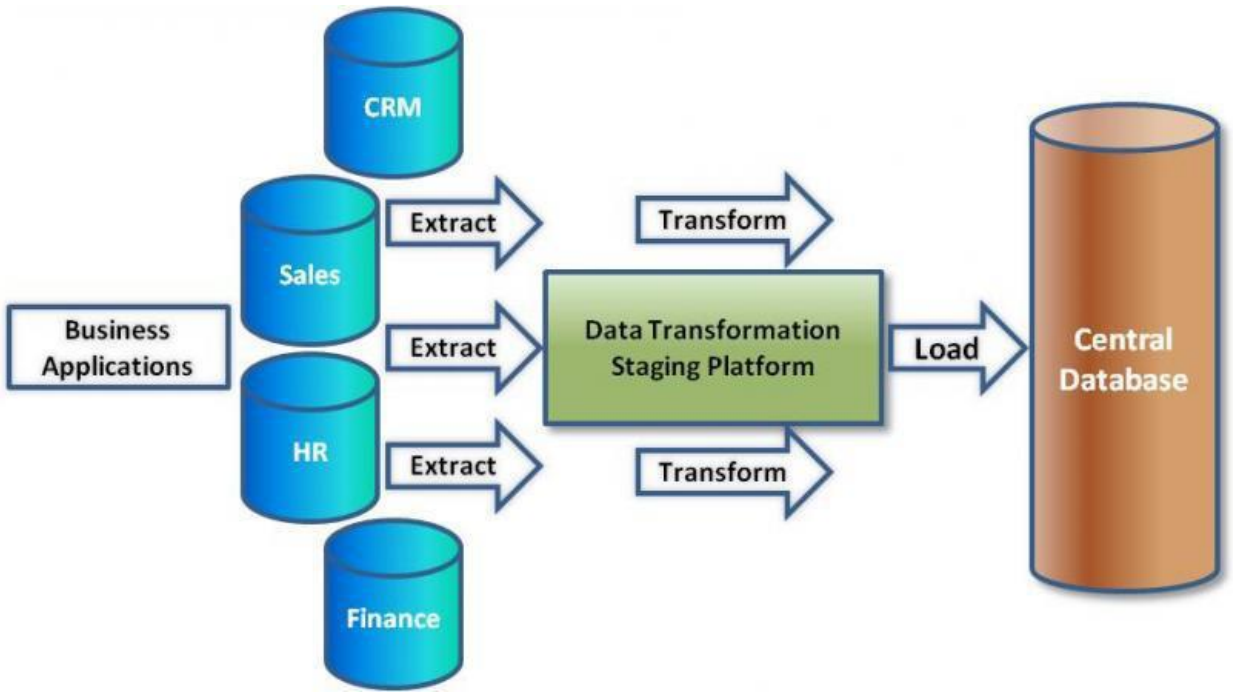

Fig. 1 ETL process [1]

ETL is a process which is used to extract data from various sources, transform that data to desired state by cleaning it and loading it to a target database. [3]

In ETL process, Data is extracted from only one data sources at a time. So ETL process becomes time consuming because DW has large amount of data. So parallel data extraction is required.

2. Extraction Methods in Data Warehouses [1]

The extraction method you should choose is highly dependent on the source system and also from the business needs in the target data warehouse environment. Very often, there is no possibility to add additional logic to the source systems to enhance an incremental extraction of data due to the performance or the increased workload of these systems. Sometimes even the customer is not allowed to add anything to an out-of-the-box application system.

2.1 Logical Extraction Methods

There are two types of logical extraction:

1. Full Extraction
2. Incremental Extraction

Full Extraction

The data is extracted completely from the source system. Because this extraction reflects all the data currently available on the source system, there's no need to keep track of changes to the data source since the last successful extraction. The source data will be provided as-is and no additional logical information (for example, timestamps) is necessary on the source site. An example for a full extraction may be an export file of a distinct table or a remote SQL statement scanning the complete source table.

Incremental Extraction

At a specific point in time, only the data that has changed since a well-defined event back in history will be extracted. This event may be the last time of extraction or a more complex business event like the last booking day of a fiscal period. To identify this delta

change there must be a possibility to identify all the changed information since this specific time event. This information can be either provided by the source data itself such as an application column, reflecting the last-changed timestamp or a change table where an appropriate additional mechanism keeps track of the changes besides the originating transactions. In most cases, using the latter method means adding extraction logic to the source system.

Many data warehouses do not use any change-capture techniques as part of the extraction process. Instead, entire tables from the source systems are extracted to the data warehouse or staging area, and these tables are compared with a previous extract from the source system to identify the changed data. This approach may not have significant impact on the source systems, but it clearly can place a considerable burden on the data warehouse processes, particularly if the data volumes are large.

Oracle's Change Data Capture (CDC) mechanism can extract and maintain such delta information.

2.2 Physical Extraction Methods [1]

Depending on the chosen logical extraction method and the capabilities and restrictions on the source side, the extracted data can be physically extracted by two mechanisms. The data can either be extracted online from the source system or from an offline structure. Such an offline structure might already exist or it might be generated by an extraction routine.

There are the following methods of physical extraction:

1. Online Extraction
2. Offline Extraction

Online Extraction

The data is extracted directly from the source system itself. The extraction process can connect directly to the source system to access the source tables themselves or to an intermediate system that stores the data in a preconfigured manner (for example, snapshot

logs or change tables). Note that the intermediate system is not necessarily physically different from the source system.

With online extractions, you need to consider whether the distributed transactions are using original source objects or prepared source objects.

Offline Extraction

The data is not extracted directly from the source system but is staged explicitly outside the original source system. The data already has an existing structure (for example, redo logs, archive logs or transportable tablespaces) or was created by an extraction routine.

You should consider the following structures:

- **Flat files** Data in a defined, generic format. Additional information about the source object is necessary for further processing.
- **Dump files** Oracle-specific format. Information about the containing objects may or may not be included, depending on the chosen utility.
- **Redo and archive logs** Information is in a special, additional dump file. Transportable tablespaces.

3. Proposed System.

In ETL process, Data warehousing systems run ETL jobs at a regular time interval, such as daily, weekly or monthly. [4] Data is extracted only from one data source at a time. So speed of data extraction is very slow. But we apply parallel data extraction than speed of data extraction is increases. For that this new method is applied to your different data source.

Here Oracle 11g and MySQL is used as different data source. Parallel data extracted from this data source and stored into Excel (.xls file). For parallel data extraction Thread, Thread Pool and Thread Executer engine concept are used. Java Technology is used for implementing this new method. Apache POI is used to create excel file at particular directory.

4 Technology for Proposed Method

4.1 Java Thread Pool

A *thread pool* is a managed collection of threads that are available to perform tasks. It represents a group of worker threads that are waiting for the job and reuse many times. Thread pools usually provide:

- Improved performance when executing large numbers of tasks due to reduce per-task invocation overhead.
- A means of bounding the resources, including threads, consumed when executing a collection of tasks.
- **Better performance** it saves time because there is no need to create new thread.
- It is used in Servlet and JSP where container creates a thread pool to process the request.

Thread Pools are useful when you need to limit the number of threads running in your application at the same time. There is a performance overhead associated with starting a new thread, and each thread is also allocated some memory for its stack etc.

4.2 Executors

Executors are capable of running asynchronous tasks and typically manage a pool of threads, so we don't have to create new threads manually. All threads of the internal pool will be reused under the hood for revenant tasks, so we can run as many concurrent tasks as we want throughout the life-cycle of our application with a single executor service.

Most of the executor implementations in java.util.concurrent use *thread pools*, which consist of *worker threads*. This kind of thread exists separately from the Runnable and Callable tasks it executes and is often used to execute multiple tasks.

Using worker threads minimizes the overhead due to thread creation. Thread objects use a significant amount of memory, and in a large-scale application, allocating and deallocating many thread objects creates a significant memory management overhead.

5 Purposed Algorithm

Step:- 1 Select multiple databases [MySQL and Oracle 11g]

Step: - 2 Make Multiple Database connectivity.

Step:- 3 Implement Executer Interface

Step:- 4 Create Thread Pool

Step: - 5 Run Executer Service.

Step: - 6 Generate Appropriate Excel Sheet.

Step: - 7 Map database table with excel sheet.

Step:- 8 Identify the window [time for data extraction]

Step: - 9 Data is ready for transformation.

Step:- 10 Stop

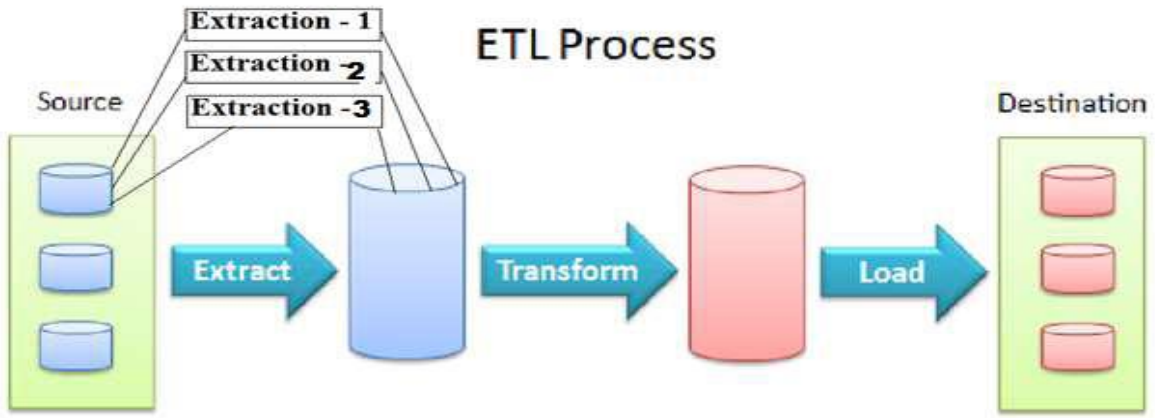

Figure: - II Parallel Data Extraction in ETL

6. CONCLUSION

Parallel data extraction is more efficient than Sequential data extraction. See Figure III. This is possible due to thread pool and thread executer service technology. In sequential data extraction there is no this type of any technology that improve the performance of the system.

In this dissertation work we have done implementation on MySQL, Oracle, and NetBeans. In MySQL and Oracle database server tool we make one database and in each database we have two table like emp and student with 50 records. Here we have four table and 200 records for experiment. For more accuracy we extract data from parallel as well as sequential.

Here in both the tables in both the database we first add 10 records and then check the result. After this process we add other 10 records and then check the result. We continue this process up to 50 records. In the result we check parallel data extraction as well as sequential data extraction. Here we store our data into separate excel file.

For 10 records Parallel data extraction take 1017 MS and Sequential data extraction take 1312 MS.
1312 MS = 100 %
1017 MS =?
(1017*100) / (1312) = 77.51
So we can say that parallel data extraction take less time and performance improved by 22.48 %.

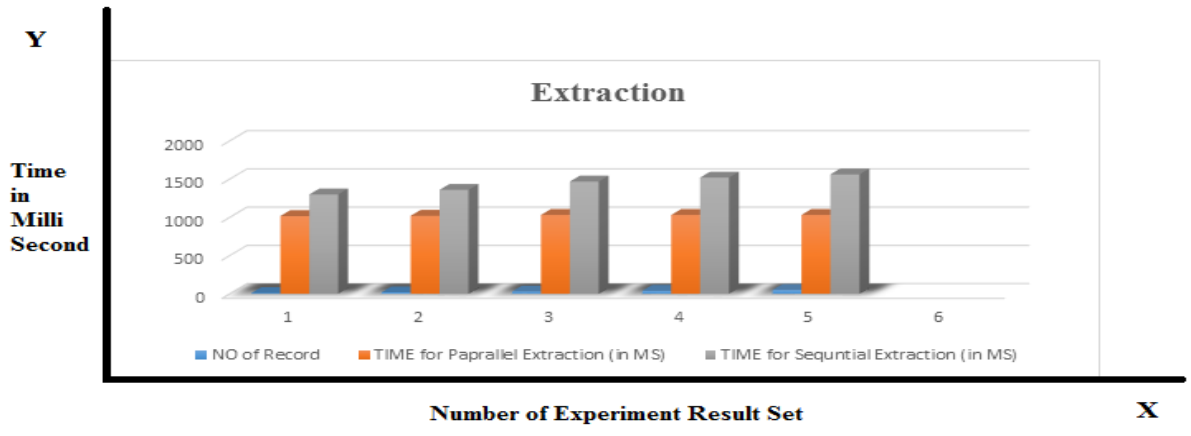

Figure: - III Comparison of Extraction Methods

7. Future Work

Data extracted from MySQL and Oracle can be stored in MogoDB or other DB. Here we stored data into excel sheet means tabular format.

8. Limitation

For this method you required very highly configured system. If your system is not highly configured than system becomes very slow.

9. References:

[1] Satkaur [1] Anuj Mehta [2] "A Review Paper on scope of ETL in retail domain." Satkaur et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(5), May - 2013, pp. 1209-1213

[2] Rajesh Yadav, Prabhakar Patil, Uday Nevase, Kalpak Trivedi, Prof. Bharati Patil "Incremental Data Migration in Multi-database Systems Using ETL Algorithm" Rajesh Yadav et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2121-2125

[3] Mr. Nilesh Mali1, Mr.SachinBojewar "A Survey of ETL Tools" International Journal of Computer Techniques -– Volume 2 Issue 5, Sep – Oct 2015 ISSN: 2394-2231

[4] Xiufeng Liu, Nadeem Iftikhar, and Per Sieverts Nielsen "Optimizing ETL by a Two-level Data Staging Method"

[5] AHMED KABIRI, DALILA CHIADMI "SURVEY ON ETL PROCESSES" Journal of Theoretical and Applied Information Technology ISSN: 1992-8645 E-ISSN: 1817-3195