==

Separate ETL Pipeline For for data transmission to train Large Machine Learning Models

Thesis/Dissertation by

Aniket Dere

In Partial Fulfillment of the Requirements

For the Degree of

Advance Research Methodologies

University of Europe for Applied Sciences

Potsdam

Contents

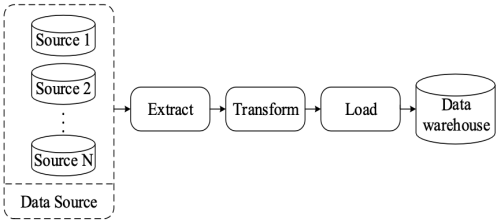
ABSTRACT

With increasing volume of data in modern world, it very hard for all industries to gather, process, transfer and find insight into the gathered data. Data collected from all the sources like sensors or RAW data generated from some devices is called Dirty data. Reason behind calling such data Dirty is because of the state of data. RAW data is sometimes unstructured and handling such data requires more efforts as such data need to clean. Unnecessary data from unstructured data should be removed. Such data is called as big data. Data after cleaning is either transform into tabular format which is later handle by structure query languages(SQL) or Unstructured query languages depending upon the data being stored into the Data warehouses. Business intelligence is built on data warehouses where data is taken to train on different algorithms. AI and Machine learning algorithms trained on this data is widely used for Management decision making.

Chapter 1

INTRODUCTION

Traditional ETL(Extract Transform and Load) method only include extracting data from source, Transforming data into tabular format and then load data into target tables with the help of software like Informatica.



Currently, Large data need to be process, for that purpose data need to process concurrently while transferring through ETL application. my research paper has suggested new approach of processing data concurrently on different thread or pipeline while transferring the same data. Pan et al. (2018) has implemented ETL tool along with SQL elements to efficiently transfer the data. He suggested the middle library in between data warehouses and data sources but it only increases the time complexity and focus on data cleaning where as we can apply another approach to find useful information in the data from data sources. My research will fill this gap by reducing the time complexity. This approach will be therefore very useful for everyone who deals with continuous generation of data. As a example we can take hospital where continuous data is getting generated for particular patient. My

approach can study and get useful information on patient once data is fed to machine learning algorithms. Data will not be stored in data warehouses. This will result in instant information on detected disease for particular patient. Similarly, we can use this approach in every possible industry to get insight into the data.

Chapter 2

PROBLEM STATEMENT

2.1 OVERVIEW

Data revolution in AI and machine learning industry has changed the volume of data getting generated through various application,sensors, actuators etc. Digital IT world need efficient and new approaches to handle such increasing amount of data. Looking for useful information inside the data is also tricky job due to large volume. To process such large amount of data new approach is discussed.

2.2 RESEARCH QUESTION

How can we train machine learning model for data generating through various resources without storing them in data warehouses? How can we reduce time required to get information from data, generating in continuous process?

Chapter 3

OBJECTIVE AND AIMS

The Objective and Aim of this paper is to achieve straight forward approach to transfer data into warehouses along with providing same data to machine learning model which will be placed on concurrent pipeline which will be running parallel with data transmission.

3.1 OVERALL OBJECTIVE

The objective of this paper is to achieve good scalability and performance of new ETL. This new approach would increase the performance of data training models as well as time complexity will be reduced significantly as new approach suggest separate pipeline for training the RAW data and separate pipeline to store data directly into data warehouses.

3.2 SPECIFIC AIM

The main aim behind whole process is to save time which is getting wasted in storing data into ware houses and again using same data to train machine learning models. however this will increase the cost of installing new pipeline and training chunks of data with regression machine learning model.

Chapter 4

LITERATURE REVIEW

According to Pan et al. (2018), We can transfer data by adding middle element called middle library along with ECL(Extract,Clean,Load) and TL(Transform and Load). Here ECL has function of cleaning dirty data which is derieved from Source of the data. Once data is cleaned and sorted same data is loaded into Middle library work. ECL work like independent data warehouse so data will extracted from different sources finely get load into middle library. Data from middle library is forwarded to TL module which has to basically transform the data and load into final data warehouse where further operations on data will do carried out. Middle library added by Pan et al. (2018) helps in stabilising the overall process of data transmisson.

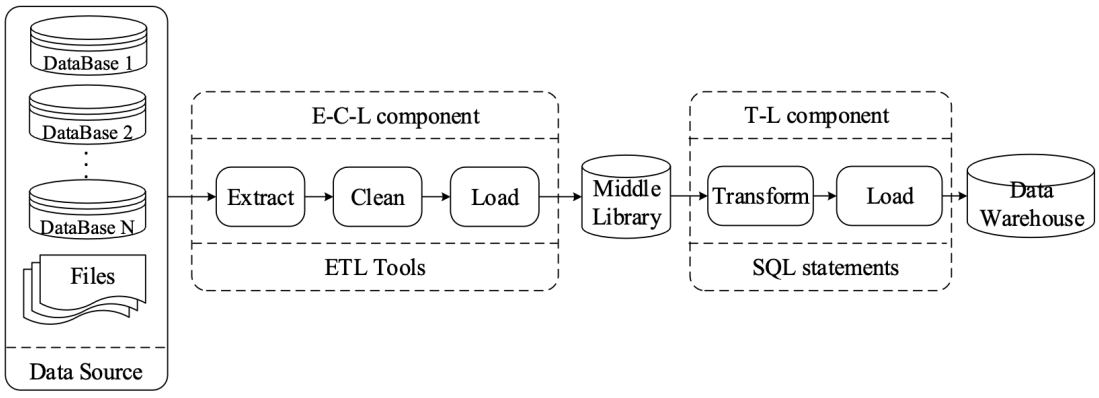


Figure 4.1: Data flow of ETL

Figure 2.1 shows how author established middle library in order to carry out data transmission. However this approach also use same traditional

technique to train machine learning algorithms. Data get stored in data warehouses and then same data is used to find insight into the data or same data is given to machine learning algorithms so that they can get trained and same model can be used in future for predicting output for new input.

Author Azaiez and Akaichi (2017),suggested a who has the focus on enhancing the performance of Extract, Transform, and Load (ETL) processes within the context of data warehousing. However,Big data often results in increased costs and performance of the transmission. The primary objective of this paper is to propose an approach aimed at augmenting the speed of ETL processes for loading into the data warehouse, with a "particular emphasis on leveraging query cache support. By implementing query caching", Azaiez and Akaichi (2017) aims to address challenges related to response time and overall performance within the data warehousing environment. The known issue in ETL, including technical issues during platform changes and integration, time and space complexity, and poor query performance, work as the motivation for this research. The proposed solution involves the strategic use of query cache, a mechanism that stores recently queried data in buffer memory, facilitating rapid execution and retrieval from the database. moreover, the author figure out the necessity of identification of difference between valid and invalid queries for Data Manipulation Language (DML) queries, with access restricted to privileged users. This research contributes insights for optimization of ETL processes, offering a valuable avenue for addressing unknown challenges and improving the overall efficiency of data warehousing systems.

Gour et al. (2010) found a groundbreaking approach to ETL processes

in the area of algorithmic trading, other than conventional methods. By using large language models (LLMs) for enhanced data analysis, this paper suggest two approaches for data collection : an Operational Data Collection (OLP DBMS) for quick data gathering and a streamlined ETL process for fast data migration and transformation into a dynamic data warehouse.

Unlike traditional ETL methods, this mentioned approach prioritizes not only efficiency and accuracy but also the specific demands of algorithmic trading algorithms. The integration of a fast ETL process aims to significantly reduce human error while optimizing the performance of trading models. The study places a distinct emphasis on the strategic use of ETL functions, emphasizing their pivotal role in data cleansing, extraction, transformation, and integration, all geared toward facilitating seamless query processing essential for algorithmic trading strategies.

Theodorou and Diamantopoulos (2019) on the other hand introduces a novel framework that extends the concept of Data Lakes to the Edge, employing a distinctive architecture for Extract-Load-Transform (ELT) processes on edge gateways. Inspired by the model of Data Lakes in conventional data centers, the notion of Data Lagoons at the Edge is proposed, serving as lightweight repositories to dissociate data ingestion from subsequent analysis tasks. The architectural design aims to bolster deployment flexibility and enable the dynamic scaling of intelligence in resource-constrained edge environments. The approach demonstrates potential in meeting diverse requirements, particularly emphasizing deployment adaptability. The ongoing research underscores the separation of data ingestion and processing, presenting a practical strategy

for efficiently scaling intelligence at the Edge. Future endeavors involve expanding the IoT prototype and conducting comprehensive benchmarking, including key performance indicators such as task setup time, migration time, and end-to-end latency relative to computational resource utilization.

Ebadifard et al. (2023) studied old approaches to Algorithmic Trading data collection and transformation into a Data Warehouse (DW). which gave significant contributions: automating data collection in a cloud-based OLTP database and implementing an ETL process for data transformation in a DBMS, enhancing Machine Learning (ML) performance in Algorithmic Trading. Notably, the paper finds differences itself through diverse technologies in each prototype and outlines future plans for scalability, including new approach to PostgreSQL OLTP and DW and using Cloud Computing and GPU resources.

Chapter 5

RESEARCH DESIGN AND METHODS

5.1 OVERVIEW

In methodology of our new propose ETL design, I have arranged new technique to extract data then transform same data and then creating 2 pipelines for that data as shown in below figure. There are some alternatives presented in other research papers but unfortunately non of them consider any approach to train data generated through different modules on live streaming. One approach suggested by Pan et al. (2018) includes adding middle library where data is stored intermediately but he is again transforming the data in order to train the data. however my proposal fits best when an organization has means to build infrastructure and need to get output of the data as soon as possible. When any organization need insight into their data then this approach is best instead of waiting for getting data stored on data warehouses and then extracting that same data from data warehouses to get useful information out of that data or train machine learning model on that data. example for this we can consider financial reports at the end of every month and quarter which get published by CEO and CFO of the company. In ideal condition data would be given to the data warehouses through either DBlinks or kafka ETLs directly. If we feed same data to machine learning model which we place on separate thread can

be then trained with capacity and CEO and CFO doesn't have to wait for sometime to declare the result of company. Medical sector also has lots of opportunities for this model.

5.2 POPULATION AND STUDY SAMPLE

In this study cases, I have considers data from diverse locations of relational and non relational databases, flat datasets for example data generated through ERP(enterprise resource planning) modules of multinational company. This datasets might vary in data structure and formats but it must be handle by proposed solution. Sometimes data is composed of structured data or semi structured data. Data from various sectors like finance, medical sectors, farming, operations etc which again mostly on research objective. Data generated from Oracle ERP model is main data considered here because of the clean data and continuous generation of data from Oracle ERP. This data is consisting of financial data from different models of Oracle ERP like General ledger, Account payable and account receivables.

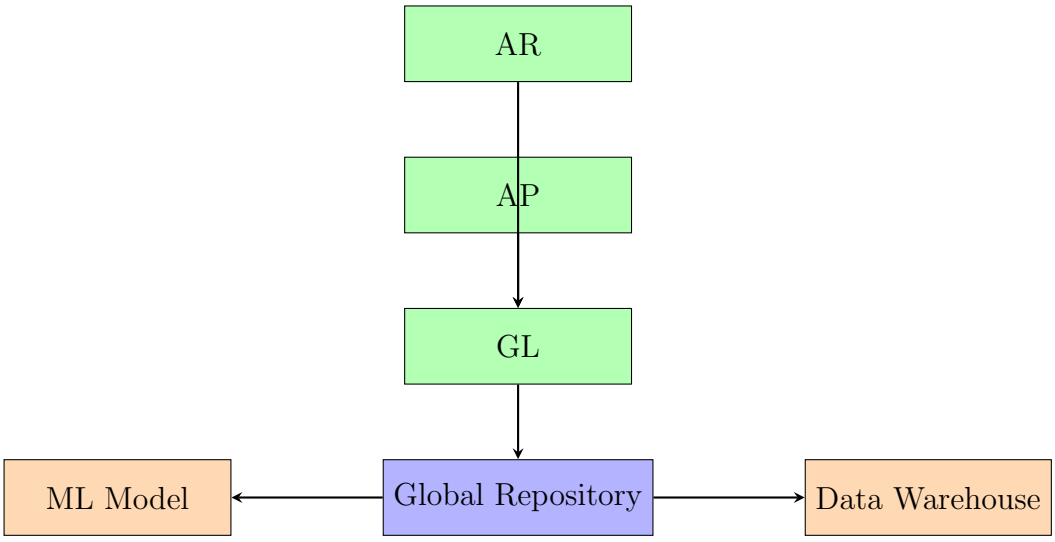
5.3 SAMPLE SIZE AND SELECTION OF SAMPLE

Data generated from Oracle ERP is huge and mainly consist of assets, payslip and all details about inward and outward financial details. Sample size which is considered for analysis is 100k rows generated from general ledger. We are considering only to transfer only 1o percent of original data volume as we will be testing the model in regression pattern. data will be continuously feed to machine learning model placed

on concurrent thread. Therefore data will be transfered to data model in chunks.

5.4 SOURCE OF DATA:

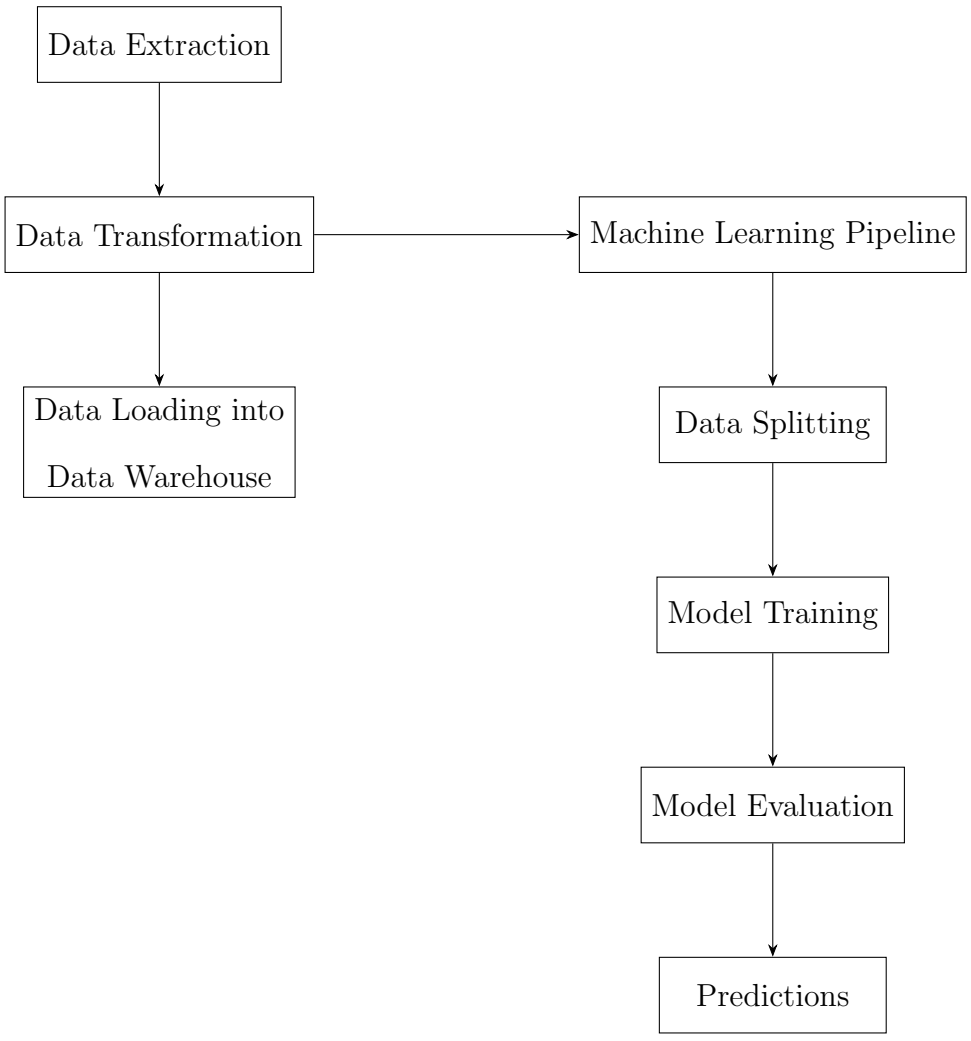
Data source is consisting of the data from Assets own by company, Payslips of employees etc. All the data is generated through account payable(payment which is paid by organization to customers or employees), account Receivables(payments which are received by company). Data generated from these module is deposited into respective tables. Data is then flown to general ledgers books tables with more information.



General ledgers has tables which consists data combination from different sources in source and category wise distribution. I considered GL JE LINES table for data transmission purpose. data get generated and stored as shown in fig. Table has columns name- journalName, headerId. These columns help in identifying each row uniquely. Here data is directly transferred from general ledger to global repository as both of them resides in the same database.

5.5 DESIGN

Implementing proposed model is very easy if organization is already using tools like informatica, Kafka for data transmission purpose. Below data flow diagram explain how data get flow from data sources and predictions or useful information is extracted from the data received. while everything is getting processed on parallel thread.



Data extracted from ERP module or sensors is in RAW format therefore need to clean before transferring same data to data ware houses or machine learning pipeline therefore at stage 2 data get transform into any format user desires but mostly tabular format. Traditional model here can be redesigned as attaching machine learning model to data

warehouse as it will wait for all the data to load into warehouse and then do the data spliting and model training based on that data. Proposed solution design implements a separate pipeline here which takes accept data into chunks. Data is then processed through various steps as shown in diagram. data is splitted and then model can be trained on that same data. This model is what we call result and using model we can evaluate any sample data and predict our predictions based on it. Therefore proposed model serves better for real time data processing and building models or making predictions real time.

Chapter 6

EXPECTED RESEARCH FINDINGS:

For study purpose, we used data generated by Oracle ERP module. This data passrd through new pipeline should be able to detect the financial revenue and insights of this data. Data passed is huge in volume therefore load balancing and infrastructure should bear the load.

- **Performance Metrics:** Processing time for transferring data from source to destination should be similar to the time taken for usual ETL.
- **Modularity and flexibility:** delivered pipeline should be easily used for different data cases and data environment. also modularity should focus on the re usability of pipeline for different scenarios.
- **Decoupling success** Model should be the success of decoupling data ingestion from processing and should have adaptability.
- **Adaptability of changes:** Data received through various sources can be vary in the various formats like linear data, Structured data, unstructured data. These different kind of data should be handle equally. Designed pipeline should adopt to any changes made in the data format. Sometimes primary key and other columns might get different formats of data irrespective of the current table column data type. This kind of adaptability should be taken care of by it.
- **Real time monitoring:** Data can be monitor real time which is the real motive behind this model. Data should get feed to machine learning model

placed on another thread. This data is suppose to generate report or train a model based on data, it is receiving through pipeline. trained model should be efficient and can be trained further depending upon organization need.

Chapter 7

STRENGTH AND WEAKNESS OF THE STUDY

Strength

- **Reduced Time :** Time required to stored data into ware houses and again retrieving same data from data warehouses it cut to directly processing same data on real time parallel threading process, resulting into decreasing the time complexity from 3n to n.
- **Data Integration:** Data from diverse data sources can be integrated in this pipeline resulting into low dependency on unnecessary infrastructure.
- **Consistency and Accuracy:** Data will be flowing continuously into parallel thread which result into training machine learning models in regression pattern. Thus increases the accuracy of the training model.
- **Scalability:** Model can be always expand depend upon requirement of data to be processed. Initially I have considered only one thread to process data which can be expanded to no of the threads depend upon algorithms company is considering for training.
- **Data preservation:** Data is always getting stored into data warehouse which make sure data is safe and all historical data is preserved.

Weakness

- **Real time latency:** Sometimes data generated through sources might have latency which might resulting into wrong input to the machine learning model.
- **Resource Intensive:** Data volume also determines how resources will be planned which might result into unnecessary resource allocation if low volume is getting generated and vice versa.
- **Cost of model:** Implementing different pipelines for algorithms will surely increase the cost and can be affordable by large organizations.
- **Complications in implementation:** Implementing new pipeline with new features will increase complications along with overall load on databases.

Chapter 8

CONCLUDING REMARKS

In conclusion of new proposed ETL development it represent a significant straight forward in addressing the increasing challenges of data processing in software industry. Our model, designed to decouple data ingestion from processing, outline major improvements in time complexity and adaptability. The results of our performance evaluations demonstrated enhanced scalability, modularity, and real-time adaptability, noticing the model's applicability in dynamic data sources.

REFERENCES

Noura Azaiez and Jalel Akaichi. Override traditional decision support systems-how trajectory elt processes modeling improves decision making? In *International Conference on Model-Driven Engineering and Software Development*, volume 2, pages 550–555. SCITEPRESS, 2017.

Nassi Ebadifard, Ajitesh Parihar, Youry Khmelevsky, Gaetan Hains, Albert Wong, and Frank Zhang. Data extraction, transformation, and loading process automation for algorithmic trading machine learning modelling and performance optimization. *arXiv preprint arXiv:2312.12774*, 2023.

Vishal Gour, SS Sarangdevot, Govind Singh Tanwar, and Anand Sharma. Improve performance of extract, transform and load (etl) in data warehouse. *International Journal on Computer Science and Engineering*, 2(3):786–789, 2010.

Bin Pan, Guomin Zhang, and Xuepei Qin. Design and realization of an etl method in business intelligence project. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 275–279, 2018. 10.1109/ICCCBDA.2018.8386526.

Vasileios Theodorou and Nikos Diamantopoulos. Glt: Edge gateway elt for data-driven intelligence placement. In *2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE)*, pages 24–27. IEEE, 2019.

APPENDICES

A.1 CITATION MANAGEMENT SYSTEM

I have used bibtext format of latex to write this paper for citation management, where I am citing all the paper from google scholar itself and using its bibtext in separate file. Screen shot of the file is attached(Figure A.1).

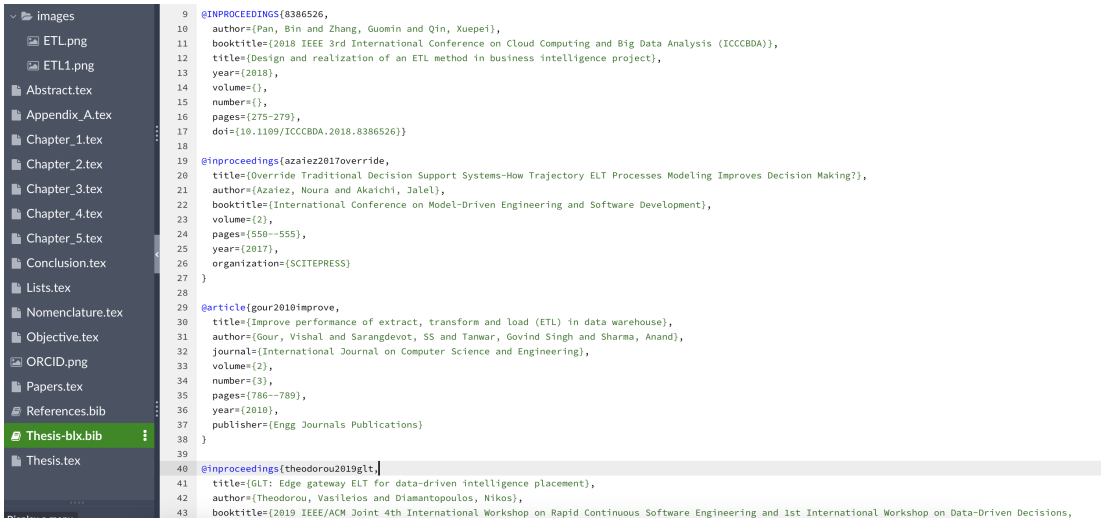


Figure A.1: bibtex Citation management