National College of Ireland

Enhancing Extraction in ETL flow by modifying as P-ECTL based on Spark Model

MSc Research Project
Msc in Cloud Computing

Rahul Bagave
Student ID: x18147038

School of Computing
National College of Ireland

Supervisor: Sachin Sharma

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Rahul Bagave
Student ID:	x18147038
Programme:	Msc in Cloud Computing
Year:	2018
Module:	MSc Research Project
Supervisor:	Sachin Sharma
Submission Due Date:	20/12/2018
Project Title:	Enhancing Extraction in ETL flow by modifying as P-ECTL based on Spark Model
Word Count:	5938
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	27th January 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Extraction in ETL flow by modifying as P-ECTL based on Spark Model

Rahul Bagave
x18147038

Abstract

Big data is a term that refers to a massive volume of data that cannot be stored or processed efficiently with a traditional Extract Transform Load (ETL) system within a given time frame. The amount of data generated in enterprises these days is too huge and may surpass the current processing capacity. The processing of this big data with the existing ETL methods can be a tedious job as it consumes an additional amount of time. Presently, standalone system is used to extract data from varied data sources. The proposed approach deals with the current problem of processing big data for the analytical phase, precisely in Business intelligence. To solve the issue of processing massive data, the proposed framework will extract data with spark framework and distribute the extraction task over different nodes. Also, removal of dirty data takes place at the initial level, and extraction tasks running on different machines is serialized and dumped in the Relational Database Management System (RDBMS) using the Sqoop framework. The primary purpose of this research focuses on enhancing the traditional ETL process. This can be done by implementing the proposed Parallel-Extract Clean Transform Load (P-ECTL) framework based on the Spark Model. The test result shows that the P-ECTL framework improves the prolonged Extraction process and stack up well against current extraction techniques.

Keywords- Hadoop,ETL,Spark,Business Intelligence

Contents

1 Introduction

Business Intelligence is a system wesshich comprises of raw data collected from applications, processes, set of methods and transform them into meaningful information that can be used for strategies and tactical decision making. Business inclines towards the application domain, whereas Intelligence represents resources and techniques used for the successful running of an enterprise.Business Intelligence comprises of many processes such as ETL, visualization, reporting,etc.Data used in Business Intelligence tools is stored in Datawarehouse and it is extracted using an ETL tool [1].

ETL refers to Extract - Transform - Load, which plays a crucial part in a data warehouse that involves extracting data from varied sources, with an interim step to transform it as per the operation and then load into the end target [2]. The initial step in this process is to extract data from the source system. Extraction is the most crucial phase in ETL because the subsequent methods depend on the data obtained from the source. The extraction process aims to convert data into the desired form so that it can be a useful resource for further operations. The transformation phase performs a series of changes in raw data so that data is suitable for the end target. Data Cleaning is an essential phase as it aims to pass cleaned data to the desired destination. Some examples of data cleaning includes filtering columns, translating values, joining data from various sources, pivoting, sorting data, transforming values based on flags, etc. Lastly, the load phase dumps the refined data into the end target, usually a Datawarehouse. Loading data into the data warehouse depends on the requirement of the enterprise. Some data warehouses may require data to be updated daily, or some may need to overwrite the existing data on a periodical basis.

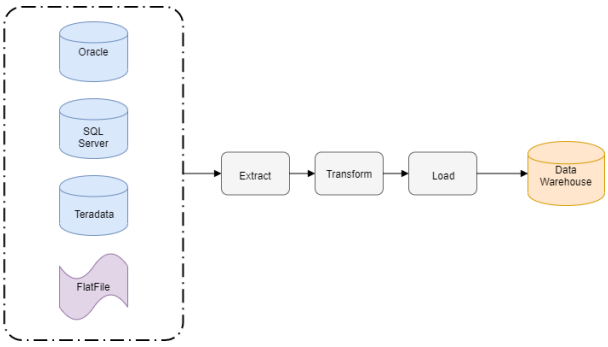

Figure 1: Traditional ETL Flow

As shown in figure 1, data is pulled out from the source system and stored in the destination. Source systems can be of different types, such as text files, CSV files, databases, media, etc. Data extracted, could be of various sizes and cannot be directly used in its form. Hence, it is necessary to transform and load.Corrupt and inaccurate data usually is termed as dirty data. It is essential to clean this data and correct values for known entities. For cleaning this data thoroughly, an enhanced method known as ECL-TL was introduced. This approach separates the traditional ETL approach into two-part known as ECL and TL, with the introduction of the middle library [3]. The modified approach cleanses all dirty data, which in turn increases the flexibility and efficiency of the system. Big data plays a vital role in all organizations because it enables them to collect, store, manipulate massive amounts of data at the correct speed, at the right time, to gain proper insight.

One of the biggest challenge faced by organizations today is to process tremendous volume of data. To process large volume of data, making use of spark over the other frameworks can be advantageous. It processes data by splitting petabytes of data into smaller chunks and then processing them on the various clusters parallelly. In our paper, a modified version of the ETL framework will be used to extract data with the use of spark and changing the framework of P-ECTL. Data will be pulled off in parallel from the source system, that will consume less time for extraction, which, in turn, will optimize the extraction process and hence increase the overall efficiency of the ETL workflow. The process starts extracting data from the different source systems using spark on multi node cluster and store it in HDFS(Hadoop Distributed File System). This system gets maximum efficiency when spark runs with HDFS [4].Extraction with traditional methods can be time-consuming when the source system is in terabytes or petabytes, so-called Big Data. Sometimes due to long waiting time of query execution, the process can fail and lead to an incomplete transfer of data from source to the staging area. Failure in the extraction process forces the user to start the process from the start. Re-iterating the method is not cost-efficient and can result in failure of the dependent process.

The following paper gives a summary of the research conducted, and the system designed for the same.Section 2 outlines the background and the relevant work in the same field.ie. ETL and Bigdata. Part 3 demonstrates a methodology that has been followed with consists of the design overview and design flow. The proposed Implementation has been described in section 4, which consists of system configuration and data set summary information. Following the implementation part, several tests are carried out, which are explained in segment 5. The paper ends with a conclusion and references used in section 6.

2 Background

This part of the section explains about the background work done by modifying the traditional ETL framework. In an attempt to enhance the performance of the data extraction process, this section underlines the principle approach practiced to improve the performance of the ETL system by dividing the framework into two parts: ECL(Extract-clean-Load) and TL(Transform -Load). Further, the research paper presents a novel approach to load data from the source system to the destination from disparate source systems. ETL provides easy GUI to where users can create packages performing multiple extraction tasks.

The ETL process is accomplished by modifying the traditional approach highlighting the below shown in figure 2 with stages involving steps such as extraction,clean and Load in the first stage. Further, the data is loaded into the staging directory after cleaning data [3]. Finally, it underwent transformation and loading using SQL statements.

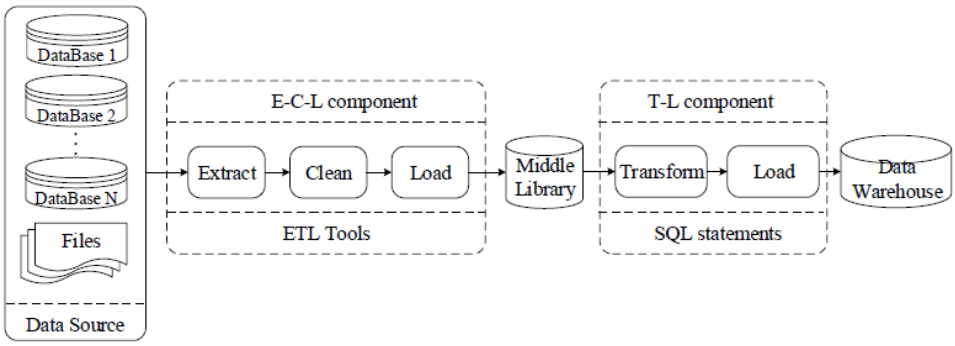

Figure 2: ECL-TL Flow [3].

Data Source: The data Source consists of various types of database systems.The most common data types are Databases, Various file types, and live feeds. The most common RDBMS databases are oracle,MySql,sql server ,DB2,AmazonDB and so on.
ECL: The ECL stands for Extract, Clean, and Load, which is connected to different data sources. The extract is used to gather the raw data from various sources and passed for cleaning. After passing it through data clean, it is made sure that it is free from dirty data.
Middle Library: The introduction of the middle library in between ECL and TL provides data to get dumped and makes it easily accessible for further processes. The middle library aims to store the cleaned data received from varied clusters.
TL: The data received from the middle library can be transformed and loaded into a data warehouse as per the business requirement. This data is crucial and hence, is stored on their infrastructure. Since the middle library is locally placed, it is easy and less time consuming to extract and manipulate the data.

2.1 Related Work

The initial contribution in the field on ETL started with implementation in ARKTOS 2. ARKTOS 2 is a design tool for ETL, which is responsible for the identification of data from the source side. The author in [5] states that ,system extracts the information coming from varied sources with customizing and then integrating information to propagate data

to the data warehouse or data mart. The goal of this project was to optimize and design the data flow at the initial stage.The author has proposed the use of Unified modelling language for conceptually modeling the flow of ETL. The paper describes more about the approach taken to ease the operational task in ETL, such as the integration of different sources, the transformation sources, and the destination attributes. Furthermore, the author describes the mechanism of ETL, and the large ETL processes can be easily made into a simple package [6].

Thomsen [7]Introduced a pygraetl framework where users can use in python language for the development of ETL packages. With this, the developers could efficiently process data using inbuilt functionalities using data access for fact and dimension tables. This novel approach was introduced with an intention to reduce both the execution and developing time. The performance was better than GUI tools but was meant to be used only by python users, which limits the number of users.

The authors proposed a method to integrate various data sources and implementation of analytical methods. The process proposed in [8], describes the combination of multiple data sources and the implementation of the systematic method. Incoming data is heterogeneous and can have several sources. The main problem of data integration in the extraction method has to be resolved so that a suitable analytical approach can be implemented. This is called knowledge discovery from the data process. ETL can solve the problem of data integration, where is routing in different formats. ETL tool facilitates flexible management of the ETL process for data integration. This job can be automated to perform the daily task and is adaptable to manage data integration. Adding a path to the data source can be quiet difficult with wrapper class and need programming principles.

A semantic ETL framework is explained in [9],which is used to combine and distribute data from varied sources. The above study comprises of analysis if data manually, ontology engineering that needs a good understanding. This is a time-consuming process as it is a manual process and requires more time consumption. Hence a big data can be processed but with long-time execution of the process.

The challenge to process a large volume of data quickly and efficiently using Mapreduce is proposed in [10]. The paper illustrates the use of ETL using MR to load data into the data warehouse.The author has named it an ETLMR framework and supports both star and snow schemas. In this article, the author has used map-reduce, which provides high-level ETL specifics on data. After integration with MapReduce, it provides data synchronization with nodes. Data can be processed in multiple methods, including processing by single tasks or by all functions present. In this case, the data is partitioned into equal sizes, and tasks are processed in parallel. The proposed approach can reduce time-consuming up to specific time, but still, Hadoop writes data to disk and then processes tasks parallelly.It will surely increase the efficiency, but since there is a use of Map-reduce, the data will be stored in Hdfs on Hadoop, which is not as good as using a spark that stores data in memory

Similar to the problems mentioned above, Bansal [11] found that big data processing with traditional ETL systems is difficult. The author proposed an ETL framework that uses a semantic method to integrate heterogeneous data. The model discovered offers basic integration and knowledge from multiple sources. Data is created using Resource Description Framework(RDF) as a graph data model, and the extraction of beneficial information is done using SPARQL. The experiment is done using the dataset having information on fuel and transportation economy. The author failed to show results with

this approach, and therefore there is no definite idea or the impact factor of the same.

Before looking into the further process and frameworks to process big data,this [12] paper explains the overview of HDFS (Hadoop Distributed File System), Master and worker node cluster environment, MapReduce, Hadoop Framework. The author explains the stack structure for Hadoop around the significant data components such as Hadoop Core Framework, MapReduce, HDFS, PigLatin, HiveQL, Sqoop, and hive. HDFS and MapReduce are the Low-level components that retrieve the analysis results from Hadoop Core Framework. The High-level components consist of PigLatin, HiveQL, Sqoop, and hive, which creates a developer-friendly framework to perform analysis on the data in the Bigdata cluster. In a precise way, the data model is components in detail and the means to use these components in a user-friendly manner in the real-world.

Samira [13] explains how a vast amount of big data can be processed with Map reduce on multiple clusters. The experiment performed uses amazon web services to setup Hadoop Mapreduce for a multinode cluster. Dataset of 1 GB text file is taken and processed using Map reduce concept. The article does not explain different scenarios with an increase in data size. Also, Mapreduce has a limitation to process only batch files where the spark can be used for both batch files and live streaming data, which is much faster as compared to MapReduce.In house cluster management system is responsible for map-reduce implementation and distributing and running of tasks. Cluster management handled by the author in this paper is in spirit to other system like [14].

Still there is no efficient way to extract big data to enhance the process.Soon a new framework was proposed in [15]which makes use of sqoop and Hadoop to load data from rdbms to Hadoop and vice versa.The author here is using sqoop to load data from RDBMS and vice versa.The author has made use of oozie to rub job at a given schedule.But with Hadoop system, we can extract the data with parallel computing where hadoop is mainly used for batch processing .Author here suggest to use spark because of its in memory processing and mostly used to real time data processing.

There is an another approach with parallel processing where author suggest to parallelize ETL framework with CloudETL which uses Hadoop and Hive as a warehouse system to dump data into database.There has been a efficient processing of updates of SCD (slowly changing dimension) data in a distributed environment. The research also explains about the type-1 SCD's in relation to mappers and executors. There has been execution of block placement policy for locating files in HDFS so that HDFS to place data to dump such that map-only job can do the work.The author suggest to enhance this process with the use of spark for faster processing [16].

3 Methodology

Going further, we will look into the design overview and design flow for the proposed system. The design overview explains the components and the architecture of P-ECTL, data partitioning in Spark, and its components whereas the design work walks through the processes in detail.

3.1 Design Overview

As per the proposed approach, further development is done by modifying the framework with the use of distributed environment concept to extract the data. This section highlights the best approach to accomplish the goal. The proposed strategy not only makes the process execute faster but also preserves the data consistency and data backup for disaster recovery.

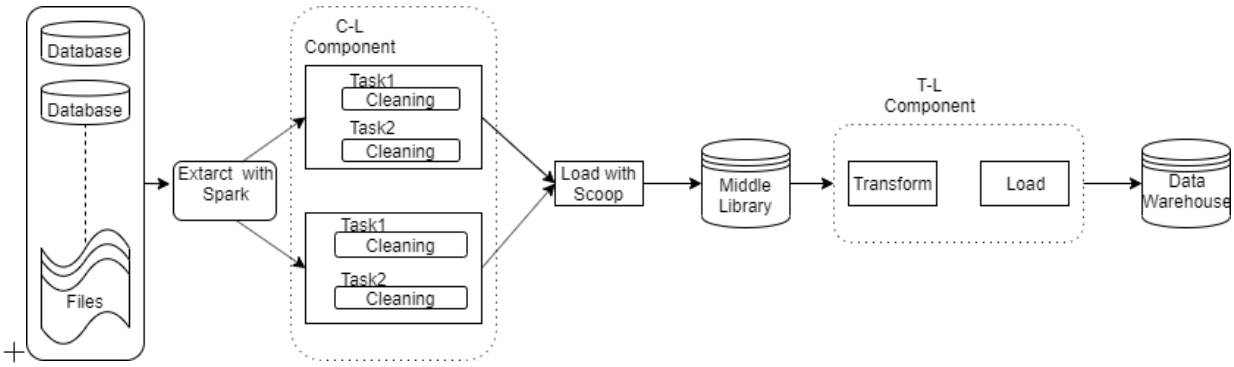

Figure 3: Modified Framework ETL as P-ECTL

As a result of our study from the given high-level detail of data extraction from different data formats, let's have a look into the core components of the system, which are necessary to understand the fundamental elements such as Datawarehouse, Middle library, ETL and the Data source. As per the above diagram 3, the proposed system architecture includes a few logical stages starting with extraction from Data source using Spark, followed by Cleaning of data in the slave node, which is then succeeded by loading the data into the middle library using sqoop, transforming the data, and loading into Dataware house.

The research further explains the influence of data getting divided and then present the extraction approach to retrieve data from the source. In addition to this, we offer the overall architecture of the P-ECTL platform.

3.1.1 Data Partitioning

Partitioning in a general means to divide the data into smaller chunks and process them on different servers. This approach of executing data on a parallel machine can enhance the extraction process. In spark, data is stored in the form of RDD's(Resilient Distributed Datasets). Since the data size is large, the data is partitioned across various nodes. Spark automatically partitions RDD's and then distributes it. The data is partitioned into key-value pairs where similar keys or range of keys are in the same partition, which minimizes shuffling, and hence, the process becomes fast. Spark works on the data locality principle, where processing nodes receive data that are nearer to them. The partitions are never

spanned across multiple machines. Spark does not allow its users to fine-tune their RDD's partitioning. Custom partitioning can be done only with key pair RDD's. RDD's partitioning can be created by providing explicit partitioner with partition by method on Rdd's or by applying a transformation that returns RDD's with specific partition. There are two types of data partitioning in spark.

- Hash partitioning

  The driver runs after the execution of main method in the program. It creates RDD's and performs various tasks upon successful execution of user code which also creates spark context. Spark shell indicates the creation of driver program and termination of driver indicates that application is over. The main task of driver is convert program into task along with scheduling task of executor.

  Hash partitioning is used to divide the data evenly across various partitions using the keys. Hash method is used to define the number of partitions in spark.

- Range partitioning

  Range partitioning is a useful technique for RDD's which have a particular order. In this, tuples with the same key will appear on the same machine as shown in the figure 4. Keys are based on a set of the order of keys and range.

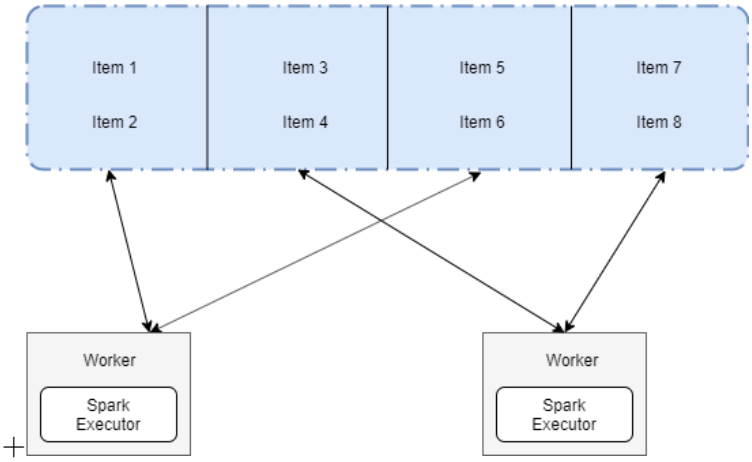

Figure 4: Data distribution in Spark

3.1.2 Spark Driver

The driver runs after the execution of main method in the program. It creates RDD's and performs various tasks upon successful execution of user code which also creates spark context. Spark shell indicates the creation of driver program and termination of driver indicates that application is over. The main task of driver is convert program into task along with scheduling task of executor.

3.1.3 Spark Context

It is the heart of Spark Engine. It is used to create RDD's, access spark services and execute jobs. It is a client of spark execution environment and acts as the master of spark.The main task of spark context is to fetching the current status of spark application, cancelling stage , cancelling job,executing job synchronously or asynchronously. [17]

3.1.4 Spark Executor

Executors are responsible for execution of each task in a given spark job. They are typically launched at the start of application. the results are sent to driver once the task runs.They facilitate with in memory storage memory for RDD's cluster that are cached by user. There play vital role in running the task that make the application and sends back the result.

3.2 Design Flow

This framework has been developed on the multi node spark framework with Hbase, and sqoop services, and the data is dumped in the Microsoft SQL Server. For further analysis and evaluation purposes, it would be beneficial if we will look around the different databases and tools used in implementing the framework.P-ETL is implemented in the spark framework and mainly consist of two modules.

- Parallel processing in Spark.
- Sqoop to parallelize the data and connect to RDBMS.

3.2.1 Parallel Processing in Spark

Spark has master-slave topology. The driver coordinates the master node, and the executor is on various worker nodes. Spark consists of the cluster manager to run spark applications and uses RRD's, which can be operated in parallel. The program code imports data from CSV to multiple nodes with the spark. Spark changes the RDD's transformations into DAG(Directed Acyclic graph) and initiates the execution. DAG converts a logical execution plan to a physical execution plan. When an activity is called, the DAG is submitted to the DAG scheduler. It divides the operations into steps, and each step consists of a unit of work known as a task. These tasks are then passed to the task manager via a cluster manager that helps to launch the application on different machines. Later, the executors execute the tasks on worker nodes. Our job has been submitted using spark-submit. In this, the main () method is called, which has been specified and launches driver program.

3.2.2 Serialization by sqoop

Post to data processing with spark and loading in Hbase, it is essential to transfer data from Hbase to the local SQL database. Apache sqoop provides interaction between the spark ecosystem and RDBMS. It can import data from SQL to Hbase and vice versa. For the data stored, Sqoop will parse the argument given in CLI and prepare the map job. Several map jobs will be launched as defined in CLI. It will divide the data, among the mapper equally to get high performance. Following this, each mapper will create a connection with an underlying database server using JDBC and write in Hbase based on the commands written.User has to pass set of parameters such connect with url, username to connect, password ,destination table name, number of mapper,Input data field delimiter [18].

4 Implementation

There are two levels in implementation, first with the Apache Spark framework on the AWS platform and the second with BIDS(Business intelligence development studio) on Microsoft Visual Studio. Analyzing the scalability of the proposed framework is done by changing the data source size and the number of tasks. The environment created is a cluster made up of three machines (nodes). It is a multi-level master-slave architecture where the master assigns the job, and slaves are the worker nodes. It is a combination of one master with two worker nodes. Each machine is 4 vCore, 16 GB memory,EBS Storage:64 GB. The device is operated by ubuntu 14.0 on the AWS EMR service. We have created a multi-node cluster on EMR and leveraged the use of the Apache spark framework on it.

In a conventional mode, BIDS(Business Intelligence Development Studio) is used for processing the data in a serialized way. The development is done in the visual studio, where the package is developed in SSIS by creating a new solution with Import/ Export wizard, which will inject one file at a time and keep on recursing the process till all the files are processed. SQL Server Management Studio 2019(SSMS) is used, with 80 GB of memory being allocated to store the data which is processed.The experiment in performed on local computer having Intel i5 chipset with two cores having 8Gb on RAM and 1Tb of hard disk. The data transferred to tables is stored in a new database dedicated for analyzing the experiment.THe data is dumped into two distinct tables having the same table structure. One of them is used to dump data with the traditional approach, and the second will be used to the proposed approach.

Since the research designs a framework that processes data in parallel, thus to perform data processed is analyzed by using 7Gb of data having 9 CSV files. The CSV files are downloaded from significant data sources available on the internet which is treated as a live streaming flight data. The data consists of flight delay and cancellation from North America region for a period of 9 years which keeps of updating as soon as there is update in the current situation. There are approximately 20.5 millions rows and 28 columns of data which includes carrier details.The final outcome from data analyzed are graphs showing frequently cancelled and delayed flights.With this result, users can get the benefit of choosing the carrier of their choice intelligently.Job submitted on spark in written python code, which recursively executes the process.Many languages can be used for this but we have chosen python due to its design and standard libraries.

Amazon Web Services Used .

There are more than 180 services provided by amazon on its cloud console platform over the internet. They provide developers with various services for machine learning, data analytics, web development tools, native infrastructure required on pay as you go model.Here , we are using AWS EMR, RDS,S3 for deploying our framework.

1. AWS EMR

   AWS EMR is used to process a vast amount of data with multiple nodes cluster.EMR comes with pre-installed spark, sqoop, hdfs by setting up a multinode cluster with the below-mentioned steps.

   - Created S3 bucket to store csv files.
   - Created a unique keypair in Network and security and download on local as .ppk

- Configured new EMR cluster with applications such as sqoop, and number of master slave nodes required.
- Enter log folder s3 location.
- Assigned the instance with key for security and access and press create cluster.
- EMR cluster launched with
- Download putty gen and ssh with the help of .ppk key.

2. AWS RDS

   It is a relational database management service that helps the user to set up a database on the cloud that can be exposed to SQL client.

   - Select RDS service from the list of AWS services
   - Create a database with the standard version for the SQL server as required and provide the region.
   - The production environment is selected as it gives high availability and consistent performance.
   - Setup DB cluster with credentials and memory size as db.t2medium.
   - The DB cluster created can be connected to SQL server client with the endpoint "database-1.cjhyypbne4x8.eu-west-1.rds.amazonaws.com" and port no.1433

Traditional ETL Approach on Visual Studio. In this part, we are going to make use of visual studio as a tool to develop package a ETL package. The package developed extracts same amount of data in serialized manner , by iterating for loop over 9 files for dumping data into SQL server for 9 consecutive times.The package has been developed from scratch by installing SQL Server Integration Services Projects Extension which provides with wizards to develop tool in BIDS(business intelligence development studio)framework.Steps followed to create package are mentioned below.

- Created destination table in SQL server as per the file structure .
- Created database connection with Sql server connection manager.
- Configure connection with flat file connection manager
- Pointed source files to destination table using data flow wizard.This is placed in for loop container which is been configured to execute for all the files present in the folder.

5 Evaluation

In the below section, we analyze performance evaluating on basis of Data Transmission Latency,Accuracy of data required for both traditional and proposed framework.

5.1 Data Transmission Latency

Data transmission latency: This metrics is the total time required for data transfer from source to destination by processing data in parallel.The data size is plotted against time vs Data size. It can be clearly seen in figure 5 that the proposed framework consumes less time. As the data size increases gradually , there is significant difference which can minimize the time consumption for extract and hence can enhance the system as proposed.

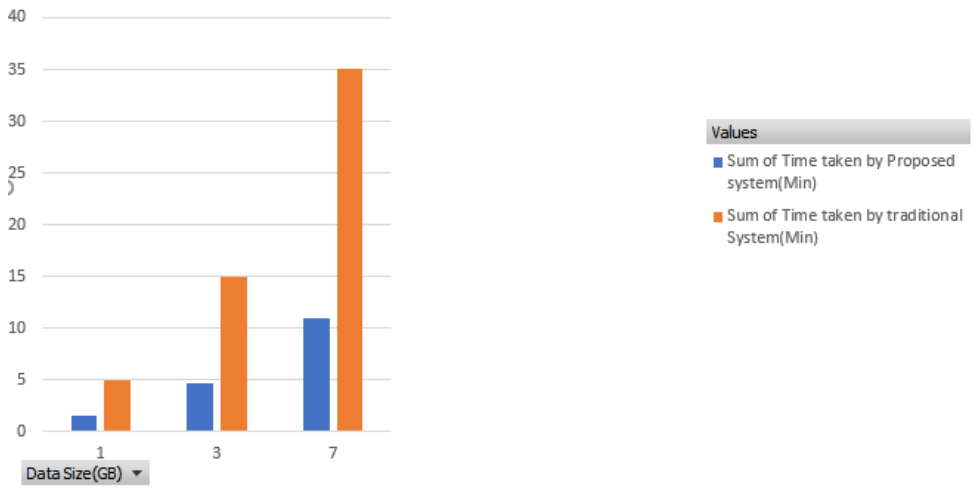

Figure 5: Graphical Representation of time vs Data Size

5.2 Accuracy in data

Accuracy can be measured by comparing the number of rows processed.Finally, the row count should should be similar, once the data has been processed under traditional and P-ECTL framework .Here, we have processed the data in both frameworks with variable data size.As it can be seen that,as the data size increases, the number of rows increases.After successful processing of data under both the frameworks, the count of rows remains same which shows that there is no data loss. .

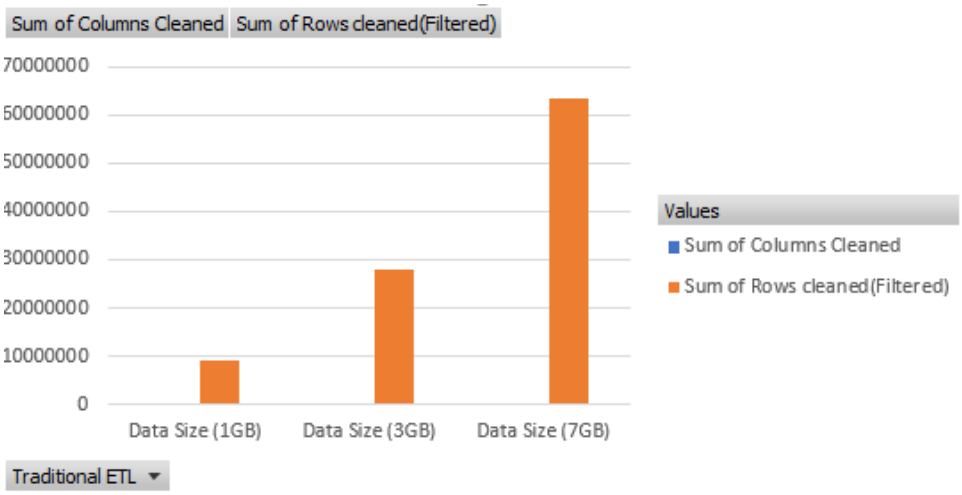

Figure 6: Traditional ECTL cleaning data

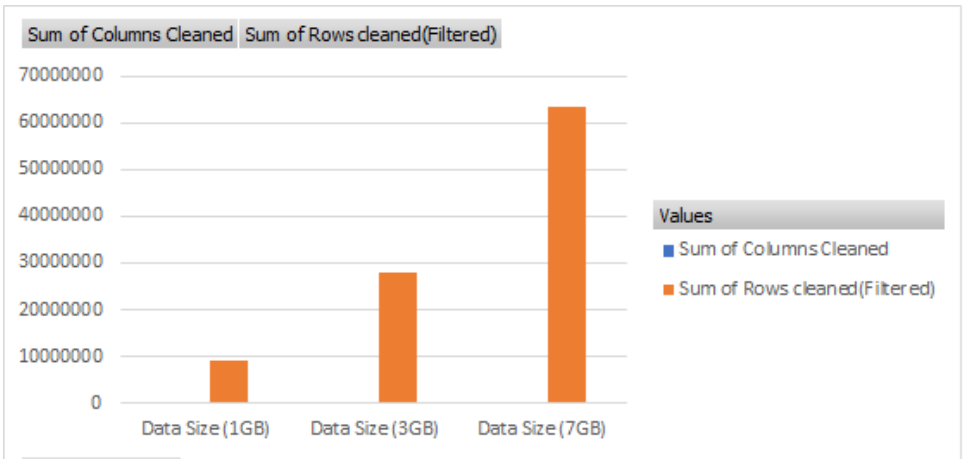

Figure 7: P-ECTL cleaning data with spark

6 Conclusion and Future Work

In this article, we started with analyzing the current ETL problem of extraction that persists in the initial stage. We presented the P- ECTL framework that extracts data using a spark on a multi-node cluster. Our proposed framework enhances the extraction in ETL, which provides developers to concentrate on business logic, rather than having a concern about the time - consuming process to extract data in a big data environment. By using spark for extraction and cleaning the dirty data, we have reduced the raw data and improved the efficiency of the system. Changing the framework has resulted in the system to be more efficient, which ensures a high-end graphical result or analysis as the end product. Furthermore, the flow can be much more modified where the data can be processed irrespective of the structured or unstructured data type.

References

[1] K. Salah-ddine, H. el Bousty, e. , M. Kabrane, K. Bendaoud, K. Karimi, and H. Oudani, "Investigating business intelligence in the era of big data: Concepts, benefits and challenges," *International Journal of Engineering Science*, 03 2018.

[2] Z. El Akkaoui, E. Zimanyi, J.-N. Mazón, and J. Trujillo, "A model-driven framework for etl process development," pp. 45–52, 10 2011.

[3] "Design and realization of an etl method in business intelligence project.," *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Cloud Computing and Big Data Analysis (ICCCBDA), 2018 IEEE 3rd International Conference on*, p. 275, 2018.

[4] K. Shvachko, H. Kuang, S. Radia, R. Chansler, *et al.*, "The hadoop distributed file system.,"

[5] P. Vassiliadis, A. Simitsis, and S. Skiadopoulos, "Conceptual modeling for etl processes," 12 2002.

[6] J. Trujillo and S. Luján-Mora, "A uml based approach for modeling etl processes in data warehouses," vol. 2813, pp. 307–320, 10 2003.

[7] C. Thomsen and T. Pedersen, "Pygrametl: A powerful programming framework for extract-transform-load programmers," pp. 49–56, 01 2009.

[8] M. Marek, "Integration of data from heterogeneous sources using etl technology.," *Computer Science*, no. 2, p. 109, 2014.

[9] S. Bansal and S. Kagemann, "Integrating big data: A semantic extract-transform-load framework," *Computer*, vol. 48, pp. 42–50, 03 2015.

[10] X. Liu, C. Thomsen, and T. Pedersen, "Mapreduce-based dimensional etl made easy," *Proceedings of the VLDB Endowment*, vol. 5, pp. 1882–1885, 08 2012.

[11] S. K. Bansal, "Semantic extract-transform-load framework for big data integration," 2014.

[12] S. Saminath.V, "Internals of hadoop application framework and distributed file system," 12 2015.

[13] S. Daneshyar, "Large-scale data processing using mapreduce in cloud computing environment," *International Journal on Web Service Computing*, vol. 3, pp. 1–13, 12 2012.

[14] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: The condor experience: Research articles," *Concurr. Comput. : Pract. Exper.*, vol. 17, pp. 323–356, Feb. 2005.

[15] B. Sindhuja and K. B. Chowdappa, "Transformation of data from rdbms to hdfs by using load atomizer,"

[16] X. Liu, C. Thomsen, and T. Pedersen, "Cloudetl: scalable dimensional etl for hive," *ACM International Conference Proceeding Series*, 07 2014.

[17] A. Shoro and T. Soomro, "Big data analysis: Apache spark perspective," *Global Journal of Computer Science and Technology*, vol. 15, 01 2015.

[18] L. B. B. B Nandana Kumar, "Integrating hadoop with relational databases using sqoop," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) ICACC – 2016*, 01 2016.