Evaluating the performance of ETL tool across multiple database systems

by

Prajesh Jhumkhawala

A Project Report Submitted
in
Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Computer Science

Supervised by

Dr. Rajendra K. Raj

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

May 2018
Acknowledgments

I would like to thank my advisor, Dr. Rajendra K. Raj for providing me with the opportunity to explore a new topic in the data management cluster and for always being there to guide me during the course of this project. Also, I would like to thank Dr. Joe Geigel for providing me with valuable feedback and insights on how to prepare various documents.
Abstract

Evaluating the performance of ETL tool across multiple database systems

Prajesh Jhumkhawala

Supervising Professor: Dr. Rajendra K. Raj

Over the past few years, the importance of Business Intelligence (BI) has increased. Companies are relying more and more on the insights generated by the Business Intelligence team and are investing more and more in the tools and technologies used by the Business Intelligence team. One important operation performed by the BI team is a process called ETL where the BI team brings in data from various data environments and stores them in a common place generally referred to as the data warehouse. It is important to understand the ETL tool as well as its compatibility with various database systems before making a decision. This project aims to analyze open source ETL tools with regards to its performance across various database systems. Additionally, this project also aims to understand how the ETL tools react when the size of the data changes.
Contents

Acknowledgments ................................................................. ii

Abstract ................................................................................ iii

1 Introduction ........................................................................... 1
  1.1 Background ................................................................. 1
  1.2 Motivation ................................................................. 3
  1.3 Problem Statement ...................................................... 4
  1.4 Related Work ............................................................. 4
  1.5 Hypothesis ................................................................. 6
  1.6 Roadmap ................................................................. 7

2 Design .................................................................................. 8
  2.1 Application Design .................................................... 8
  2.2 Architecture ............................................................. 8

3 Implementation ................................................................. 11
  3.1 ETL tools and database systems ................................. 11
  3.2 Dataset .......................................................................... 11
  3.3 Implementation for ETL tools .................................... 13
    3.3.1 Server Integration Services (SSIS) ......................... 13
    3.3.2 Pentaho Kettle Data Integration ......................... 14
    3.3.3 Jaspersoft ETL Talend ....................................... 14

4 Analysis ............................................................................... 18
  4.1 ETL v/s Database system ........................................... 18
  4.2 ETL tools v/s size of the database ............................... 19

5 Conclusions .......................................................................... 23
  5.1 Current Status ........................................................... 23
5.2 Future Work ................................................................. 24
5.3 Lessons Learned ......................................................... 24

Bibliography ................................................................. 26
List of Tables

3.1 Dataset Information .................................................. 12
4.1 Execution time for various databases in SSIS ...................... 19
4.2 Execution time for various databases in Pentaho .................. 20
4.3 Execution time for various databases in Talend .................... 21
4.4 Performance of SSIS vs increasing data size ...................... 21
4.5 Analysis for Pentaho vs increasing data size ..................... 22
4.6 Analysis for Talend vs increasing data size ...................... 22
List of Figures

1.1 Overview of ETL process .................................................. 2
2.1 Architecture ................................................................. 9
3.1 Tables created in SSIS .................................................... 13
3.2 Sample data in SQL Server ............................................. 14
3.3 Sample data in MongoDB ............................................... 15
3.4 Sample ETL process in SSIS ........................................... 16
3.5 Sample ETL process in Pentaho ..................................... 16
3.6 Sample ETL process in Talend ....................................... 17
Chapter 1

Introduction

Data Warehouse is a repository of strategic data from many sources gathered over a long period of time. Traditional DW operations mainly comprise of extracting data from multiple sources, transforming these data into a compatible form and finally loading them to DW schema for further analysis [10]. ETL process is the first step in data warehouse, and selecting the right tool to perform this type of data transformation is important because not all of the available open source tools can be the right one for the work involved [5]. The extract-transform-load (ETL) functions need to be incorporated into appropriate tools so that organizations can utilize these tools efficiently [10]. With the exponential increase in the ETL tools available in the market, there has been a steep increase in the number of available database systems, making it even more important to ensure that the correct combination of ETL tool and database system is being used. ETL tools also play an important role in the efficiency of the BI team since as the BI team derives insights from the data brought in by the ETL processes.

1.1 Background

The amount of data generated by companies on a daily basis is increasing at an exponential rate. Companies generate data from both internal as well as external data sources. Internal data is composed of logs about the various tasks performed by employees, call center data, customer service data. On the other hand, the external data is the data which the customer generates, for example, the location of the user, amount of hits to a website or the number
of times a particular query was executed. External and internal data differ from company to company, but the underlying fact is that the data generated is increasing by leaps and bounds. Under such circumstance, the dependence of the company tools to bring in the data from various sources is very high. The data generally move around in the company with the help of a process called ETL. ETL stands for Extract, Transform and Load. It describes the process where data is gathered from various sources (extract), transforming it to match the desired form (transformation) and importing it into a database or data warehouse (load).

We can see the overview of ETL process in Figure 1.1

![Figure 1.1: Overview of ETL process](image)

The working of each step is as follows:

- **Extract**: Extracting data from various sources such as SAP, ERP or other operational systems which are then loaded into the data-warehouse after performing transformations.

- **Transform**: Transform is an important step in ETL where the data which is extracted during the first step is then transformed so that the new data is aligned with the data already prevailing in the destination database. Transformation involves one or more of the following tasks:
  - Cleaning the data to make the columns in a standard format
- Modifying certain data
- Filtering out certain outliers
- Joining the data with existing data for a more accurate transformation.
- Determining new dimensions which are needed for analysis.

- **Load:** Once the transformation is completed, the data is then loaded into a data warehouse from where the data is used for analysis purposes.

### 1.2 Motivation

Consider the following scenario where a company X has 10 ETL processes running daily. These ETLs bring in data from various data sources as \( S_1, S_2, S_3, \ldots S_n \) where each data source can be either similar or different. These ETL tools perform various transformations on the data extracted from the above sources and load it into the company's data warehouse which consists database \( D_1 \). An important thing to keep in mind is that the destination of the database is a single database source, unlike the source database systems. Also, these ETL processes are interdependent such that \( ETL_2 \) depends on the data which \( ETL_1 \) brings and similarly \( ETL_3 \) depends on \( ETL_1 \) or \( ETL_2 \). These ETL processes are not triggered manually and are scheduled to run at a given time. When the ETL processes run, sometimes due to the size of data, there might be a possibility that \( ETL_1 \) does not bring in data before the start of \( ETL_2 \). Due to this, \( ETL_2 \) cannot bring in all the data it ideally should because \( ETL_1 \) has not brought all the data. This leads to a chain effect where all the ETLs post \( ETL_1 \) will not get in all the data because of its dependence on the previous ETLs. There are multiple teams in the working environment which depend on the data that is brought in by these ETL process for performing various analyses. These analyses enable management to make decisions which in turn affect the performance of the company and also play a major role in deciding the future tactics of the company. Thus, this project aims to help the company understand the ETL tool and its performance time in detail to prevent the unnecessary delay in data analysis.
1.3 Problem Statement

The aim of this project is to create an efficient evaluation of the ETL tools available in the market and how they respond to the change in the source and the destination database systems. Since ETL processes take up to 80% of the efforts in BI projects, a high ETL performance is vital to be able to process large amounts of data and to have an up-to-date database. There are various open source ETL tools and database systems available in the market hence making it even more important to analyze how an ETL tool will act as the database system being used.

1.4 Related Work

Over the recent period of time, there has been a lot of efforts directed towards analyzing the ETL tools with respect to the performance of those ETL tools with an increase in the size of data or how the ETL tool handles initial as well as incremental loads and in general the ease of use.

While comparing the generalized and big data BI tools, it was found that the choice between generalized and big data-specific business intelligence tools was separated by a bleak line. The result also suggested that since the generalized ETL tools have been in the market for longer then the Big Data specific ETL tools, the interface to use those tools is much more mature and can be understood easily. [11]

Another research which focused on comparing the ETL tools found out that the concept of data-warehouse was expanding beyond the traditional DW structure and the authors consider the concept of cloud DW, Big Data, and ELT as the major factors responsible for this shift. As suggested in the paper, companies are shifting to cloud data warehouse as they find it a cheaper and easy to scale option. Also, the research found out that there has been a modification in the traditional ETL process where the loading step is performed before the transformation step. This results in modifying the ETL (Extract - Transform - Load) process to ELT (Extract - Load - Transform). [10]
Another research which was focused on evaluating the efficiency of open source ETL tools found Jaspersoft Talend and Pentaho Kettle Data Integration as the most complete and mature open source tools available in the market. During the research, there were many experimental runs, where the parameters were modified during each run, the obtained diverse results in terms of the CPU utilization or the memory load was obtained. Despite the observation of advantages for Pentaho in some cases, they found Talend to have a much higher overall efficiency [9].

There has been researching focused towards determining the taxonomy of ETL tools and during this research, the research, they broadly classify the results into Unary activities where there are just one input and one output schema) and N-ary activities where there are multiple input schema and one single output schema.[12]

Another research conducted at evaluating the ETL tools found that Pentaho Kettle Data integration could perform the same functions as SQL Server Integration Services but had a much more user-friendly GUI. They also found that Kettle did not consume much of the resources in terms of the memory used or the CPU utilization. The paper further goes on to recommend that if the ETL tool is needed only for a small to medium data-warehouse then Pentaho Kettle is a good choice. [5]

Additionally, another research which aimed to emphasize on the technical aspect found out that most of the ETL tools are Eclipse-based and only a few tools support big data and web services. Another survey focusses on the issue of ETL maintenance and how that can be tackled[8]

Another survey focusses on the issue of ETL maintenance and how that can be tackled and talk about the various projects from the academic world such as SIRIUS (Supporting the Incremental Refreshment of Information Warehouses) which provides a metadata-oriented approach enabling the modeling and execution of ETL processes. Another project discussed in the paper is ARKTOS which provided primitives to capture the ETL tasks which are frequently used. The authors also talk about a Python-based framework called PYGRAMETL where they try to prove that ETL processes developed using GUI is less
efficient than the traditional version i.e. by writing and editing code. [7]

Another research takes into consideration the top nine open source ETL tools and talks about the advantages and disadvantages of those ETL tools when compared with other ETL tools and build on the research performed by Thomsen and Pedersen in 2008 [6]

1.5 Hypothesis

Consider example 1 where the ETLs are not able to bring in the complete data due to performance issues. This has a severe chain effect in a working environment and will affect the decision-making capacity of the company plus will result in an inaccurate analysis further affecting the company. The major reason for this issue is that there is not enough in-depth study done regarding the various open-source ETL tools prevailing in the market and how these ETL tools pair up with the database system. The research currently prevailing in the industry compares various ETL tools at a very high level in terms of the advantages and disadvantages of each ETL tools and how well the online support is present for the ETL tool. However, very few papers discuss the performance of ETL tools w.r.t the parameters which really matter the i.e. size of data, memory used, CPU processing speed used and treat ETL tools as an independent, standalone product. Even a smaller section of these papers takes into consideration the effect of the database system on the performance of ETL tool and how an ETL tool can either hamper or boost the ETL tool performance.

The main hypothesis of this project is that the performance of ETL tool highly dependent on the database systems from which it is extracting or loading the data. Also, the hypothesis suggests that by increasing the size of data in a controlled manner, we can gain a great amount of insight into how well the ETL tool handles the increase in the size of data and check if an ETL tool has a breaking point after which the time took increases exponentially.

Also, with the help of this project, companies will have a brief overview about how long each ETL tools takes w.r.t the size of the data the company brings in and w.r.t the source and destination database system in the picture. thus enabling the companies to make better decisions.
1.6 Roadmap

The remainder of this report provides detailed explanation about the application. Section 2 provides design of the application whereas Section 3 talks about the implementation in depth along with the dataset used. The various analysis derived from this project are discussed in section 4 and the conclusion along with the current status and future work is described in section 5.
Chapter 2

Design

2.1 Application Design

Application design for this project has focused on the simulation of the working environment of a real-life company where the ETL tools are responsible for bringing in data which has more than a million instances and hence is expected to take at least a substantial amount of time. Also, the data is separated into the dataset of increasing size so that the performance of the ETL tool can be measured in a controlled space. A few key elements to the successful implementation of this project and verification of the stated hypothesis are the ability to record the time taken right from the time the first row is pulled from the database to the time when the last row is inserted into the database. The time taken by the ETL tools is generally displayed at the end of the ETL process either in minutes or seconds. This is the time which is taken into consideration while performing the analysis.

2.2 Architecture

The architecture of the project is as seen in the Figure 2.1 and can be divided into the following section:

1. **Source Database system**: This block represents the source database system which contains the initial data to be loaded into the destination database. The source database already contains a pre-populated table which contains the complete dataset.

2. **ETL tools**: This component represents the ETL tool which is being considered for
that analysis. Even though, each ETL tool performs essentially the same function, the way each tool performs it is different since each tool has been configured using a different native language. It is the ETL tools responsibility to convert the data from the source database format to destination database format For example, if we are moving data from SQL server to MongoDB, then it is the responsibility of the ETL tool under consideration to convert data from the table-oriented structure of SQL server to the document-oriented structure of MongoDB.

3. **Destination Database system**: This is the destination database where the data will finally be loaded. It can be same as the source database or different.

4. **Time Taken** This component represents the measurement of the time taken by the ETL process for the given source and destination pair. The time is measured right from when the first row is extracted in the source database system to the point where the last row is inserted in the destination database and there is no more work left to
5. **Results and Analysis:** This is the final step is to measure the performance of the ETL tool in terms of the time it requires. This result is populated in a table and finally, when the entire table is populated, the analysis is performed on that table to generate valuable insights.

This application is designed around the ETL tools. Since each ETL tool has a different interface, there is no common design for the internal working of the ETL tools. At a very high level, the basic components of each ETL tool is a process, which is a collection of various elements such as database connections, flow statements, source and destination database, transformations, configuration properties and event flow which are generally defined with the help of the Graphical User Interface provided by each ETL tool but can also be defined with the help of writing code in whichever language the ETL tool inherently supports. This process will perform the extraction and loading of the data, along with connecting to the database which will provide the input and output source blocks which will be configured according to the need for the ETL tool.

Another factor of this project are the various database systems which are being used by the ETL tools. Since the database systems used in the project is a mixture of both SQL as well as NoSQL database, the way each database stores the data is different.
Chapter 3

Implementation

3.1 ETL tools and database systems

In order to perform the analysis, following ETL tools and database systems were used:

- **ETL tools**: Jaspersoft ETL Talend, Microsoft SQL Server Integration Services, Pentaho Kettle Integration

- **Database systems**: SQL server, MySQL, PostgreSQL and MongoDB

The above-mentioned ETL tools and database systems were selected because these ETL tools are the most popular open source ETL tools which are generally considered by others as well while analyzing the performance of ETL tools. Similarly, the database systems used in the project are highly ranked in comparison with all other database systems[1]. There are a total of 309 database systems available and of those, MySQL ranks 2, SQL Server ranks 3, PostgreSQL ranks 4 and MongoDB ranks 5. The tests were performed on an Intel Core i7 with 2.4 GHz processing power and 8 Gb RAM.

3.2 Dataset

To perform the analysis mentioned above, Citywide Payroll Data was used which is published by the Office of Payroll Administration (OPA) as a part of the NYC OpenData [3]. The dataset contains 2.19 million instances each containing 16 columns. It contains data such as the work location borough, base rate, agency start data, fiscal year, OT paid and OT data.
hours. The complete detail about the fields in the dataset and its data type is shown in table 3.1

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fiscal Year</td>
<td>Number</td>
</tr>
<tr>
<td>Agency Name</td>
<td>Varchar</td>
</tr>
<tr>
<td>Last Name</td>
<td>Varchar</td>
</tr>
<tr>
<td>First Name</td>
<td>Varchar</td>
</tr>
<tr>
<td>Mid Init</td>
<td>Char</td>
</tr>
<tr>
<td>Agency Start Date</td>
<td>DateTime</td>
</tr>
<tr>
<td>Work Location Borough</td>
<td>Varchar</td>
</tr>
<tr>
<td>Title Description</td>
<td>Varchar</td>
</tr>
<tr>
<td>Leave Status as of June 30</td>
<td>Varchar</td>
</tr>
<tr>
<td>Base Salary</td>
<td>Varchar</td>
</tr>
<tr>
<td>Pay Basis</td>
<td>Varchar</td>
</tr>
<tr>
<td>Regular Hours</td>
<td>Integer</td>
</tr>
<tr>
<td>Regular Gross Paid</td>
<td>Varchar</td>
</tr>
<tr>
<td>OT Hours</td>
<td>Varchar</td>
</tr>
<tr>
<td>Total OT Paid</td>
<td>Varchar</td>
</tr>
<tr>
<td>Total Other Pay</td>
<td>Varchar</td>
</tr>
</tbody>
</table>

The data is stored in column-based tables in SQL Server, PostgreSQL and MySQL whereas they are stored in a document-based structure in MongoDB. Initially, there was a different table created in each database of different sizes. For example, we can see the different tables created in SQL server in Figure 3.1.

Figure 3.2 shows how the data is stored in table-oriented database while figure 3.3 depicts how we have stored data in a document-oriented database. Each of these database is used with the ETL tools as mentioned in the next section.
3.3 Implementation for ETL tools

3.3.1 Server Integration Services (SSIS)

One of the most widely researched ETL tools is SQL Server Integrations Services. It is one of the major ETL tools being used in the Windows environment. ETL work in SSIS is mainly done in the form of packages where each package represents a process. SSIS does not have inherent support for databases like MongoDB. Also, the process speed for MySQL is also very slow. Due to this, while measuring the performance of SSIS w.r.t MongoDB, there was an external connector that was used. This connector was provided by Simba Technologies[4] and the use of this connector improved the performance speed to a great extent and we will see more analysis about this in the next section. Similarly, to speed up the process of analyzing ETL performance with MySQL, an external data connectivity tool was used. This project uses the Devart SSIS components[2] which assists in connecting
to MySQL using SSIS. This provides another important insight that the use of external connectors can sometimes boost the efficiency of an ETL tool and that an option like that should always be considered. A sample ETL process in SSIS is shown in figure 3.4 which depicts the movement of data from SQL server to SQL server

3.3.2 Pentaho Kettle Data Integration

Pentaho is an open source ETL tool in the field of business intelligence which provides users the ability to migrate data from source database which can be both unary or n-ary database to destination database. Pentaho has inbuilt support for all the database being used in this project and hence there were no external connectors used. A sample ETL process in Pentaho is shown in figure 3.5

3.3.3 Jaspersoft ETL Talend

Talend is another widely used ETL tool. Talend enables developing ETL tools with the help of a job designer which is an interactive method to develop ETL processes. The job designer itself is written in Java. As compared to other 2 ETL tools into consideration,
the GUI provided by Talend is less intuitive and difficult to understand. Like Pentaho, Talend also provides support for all the database systems and hence there were no external connectors used while analyzing the performance of Talend as well. A sample ETL process in Talend is shown in figure 3.6
Figure 3.4: Sample ETL process in SSIS

Figure 3.5: Sample ETL process in Pentaho
Figure 3.6: Sample ETL process in Talend
Chapter 4

Analysis

As mentioned previously, the project is divided into 2 sections. The results of each section provide a different insight into the ETL tool under consideration. The results are as described below.

4.1 ETL v/s Database system

As we see in table 4.1, when we analyze SSIS, the performance varies a lot depending on the database system into the picture. We also see that there are some surprising contradictions. For example, movement of data from MongoDB to SQL server gives the best results but the reverse is not true indicating that the in SSIS conversion from the table-oriented data structure to the document-oriented data structure is quick but the vice-versa is not. Similarly, MySQL takes the most time irrespective of the source database in consideration, leading to an overall poor performance. One thing that needs to be considered is that the results for MySQL and MongoDB were obtained after using the Devart component and Simba ODBC connector as mentioned in Section. Thus, if we dont use the components, the performance will further deteriorate.

Similarly from table 4.2 we can see that when we consider Pentaho as the ETL tool, in contrast to SSIS, movement of data from MongoDB to SQL Server takes the highest amount of time. Even though it performs better than SSIS in terms of its performance for MySQL as destination database, the performance is still poor. Pentaho offers the best performance when the database source or destination is PostgreSQL and has comparatively
overall lower efficiency when the source is MongoDB.

The final ETL tool which we considered in this project is Jaspersoft Talend and as we can see in table 4.3 its performance is equally good for MySQL and SQL Server as the source database and just like Pentaho, the performance is poor when the source database is MongoDB. The system on which this analysis was performed crashed when PostgreSQL was used as the source database and hence the analysis for PostgreSQL is not available. PostgreSQL acts as the best destination database irrespective of the source database system.

### 4.2 ETL tools v/s size of the database

There was an initial load made of 1.3 million instances followed by increments of data by 0.1 million instances eventually leading to a full-size data load of 2.2 million.

Table 4.4 represents the performance of SQL server in SSIS when the size of data is increased in a controlled manner. The slope in the figure indicates the performance of an ETL tool when the size of the data being loaded is increased. As we can see, the slope is rather gradual for SQL server w.r.t to SQL server and PostgreSQL but when we look at the same for MySQL, we see a sharp increase in the slope when the data size crosses the 2
Table 4.2: Execution time for various databases in Pentaho

Table 4.2: Execution time for various databases in Pentaho

million mark. This tells us that when the size of the data increases, the performance of the ETL tool will also deteriorate. Similarly, for MongoDB, the slope is rather a steep post the 1.6 million mark.

Consider the performance of PostgreSQL in Pentaho as shown in table 4.5. This graph justifies the previous analysis where we found the performance to be poor for MySQL across all database sources. Apart from that, the performance is rather gradual for other database systems.

When we look at the performance of MongoDB in Talend in table 4.6, we see that even though the start time is different for different database systems, the slope is rather same for all databases. From our previous analysis, we know that performance of MongoDB is poor in Talend. This graph further justifies the analysis and we see the starting point for different database systems are highly separated. From the above results, we can verify our hypothesis that the performance of ETL tool is not solely dependent on the tool itself but is the combined result of the ETL tool, the source database system, and the destination database system.
Table 4.3: Execution time for various databases in Talend

Table 4.4: Performance of SSIS vs increasing data size
Table 4.5: Analysis for Pentaho vs increasing data size

Table 4.6: Analysis for Talend vs increasing data size
Chapter 5

Conclusions

5.1 Current Status

For the purpose of this analysis, we have analyzed the performance of major open source ETL tools such as Microsoft SQL Server Integration Services (SSIS), TIBCO Jaspersoft Talend and Pentaho Kettle Data Integration. This analysis was performed with the most widely used and highly ranked open source database system prevailing in the market today. The database systems used were: SQL Server, MySQL, MongoDB, and PostgreSQL. The analysis was performed by changing the source and destination database systems and analyzing the time taken by each pair for a particular ETL tool and then once we had the complete data, an in-depth analysis was provided which is as stated in Section. From the analysis, we see that the combination of Pentaho and PostgreSQL works the best and has the highest efficiency. On the other hand, the performance of ETL tools was poor when the data was converted from a table oriented database such as MySQL, SQL Server and PostgreSQL to a document style database such as MongoDB. This tells us that even though NoSQL databases have become a huge thing in the recent past, the ETL support for such database is still poor as compared to the support available for SQL databases. This might be because of the additional time taken by the ETL tools to convert the table-oriented data to a document style data which has a more free-form data storage and is easier to scale horizontally. However, if we consider only SQL databases, we can see that the performance varies widely according to the ETL tool being used, the source and destination database systems.
SSIS has good efficiency when the source database system is SQL Server whereas PostgreSQL dominates Pentaho in terms of performance. The performance is almost similar to MySQL and SQL server in Talend.

5.2 Future Work

Even though there has been an in-depth analysis performed, further tests need to be performed to ensure the results hold true even when the database size has increased. Currently, the performance is measured for data of size 2.2 million. Further tests need to be performed where the data size is approximately more than twice the current size (4 - 5 million instances) and the results should be compared with the current results. This analysis will help us generate further analysis on how the ETL tools react to the size of the data. Another study that needs to be performed in the future is to make this analysis available for more number of database systems and ETL tools, especially NoSQL database systems so that the companies planning to migrate to NoSQL database will have an idea on how the ETL activities will react to that. Also, extending the analysis to more and more database/ETL tools will provide the companies with a broader idea about how the ETL tool will pair up with a database and have an approximate idea about how to schedule the ETL activities if they are planning to migrate their database from a SQL to a NoSQL database.

5.3 Lessons Learned

The aim of this project was to find the relation between the performance of ETL tool and the database system being used as the source and destination and observe if changing the source and destination database will have any effect on the efficiency of the ETL tools. Another part of this project was to understand how well the ETL tools adapt themselves with an increase in the size of the data. From the analysis, we have successfully found the effect of both, database systems and size of data, on the performance of the ETL tool. Another important lesson learned during this project was that the use of external connectors
(similar to those used for SSIS with MongoDB and MySQL) help improve the speed of the ETL process. Though the connectors used for this project were licensed for a trial period, these connectors can be licensed from the vendors for the price mentioned on their respective websites and used in the long run. This analysis will also provide the respective vendors of the ETL tools an overview of their ETL tools performance with other database systems. Thus extending this research to combine more ETL tools and database systems should remain the main focus.
Bibliography


