**Comparison of ETL Algorithms using**

**Open Source ETL tools**

Kristi LaVigne

Department of Computer Science University of South Carolina Upstate Spartanburg, South Carolina, USA September 14, 2022

[klavigne@email.uscupstate.edu](mailto:klavigne@email.uscupstate.edu)

**ABSTRACT**

Extract-transform-load (ETL) algorithms are a vital component in data warehousing. The ETL process consists of pulling data from various sources, cleaning the data, reforming the data, and inserting it into the data warehouse. The ETL process is integral in achieving the goals of a well performing data warehouse. Those data warehouse goals being accuracy, integrity, redundancy-free, and accessibility. There is a lack of standardization in this rapidly growing field, which brings the need to compare existing tools to gain insights in their performance and potential for refinements.

**Keywords**

ETL, Extract, Transform, Load, Data Engineering, Data Warehouse (DW), Business Intelligence (BI), Relational Database Management (RDBM)

# INTRODUCTION

Business Intelligence (BI) assists in increasing value and performance throughout the company. In order to access the benefits of BI, the data warehouse requires uniformity and accuracy. To achieve the goals of uniformity and accuracy, a data engineer uses the process of extract, transform, load (ETL). The extract phase is the gathering of data from various sources in various types. The transform phase involves modifying the data to conform to the data warehouse quality and standardization. The load phase is the importing of the data into the data warehouse, typically to a staging table. The data will be moved from the staging table to its production table once the quality is assured. The ETL process takes up approximately 80% of the process in achieving Business Intelligence. [1] Surprisingly, there is a lack of research on the ETL process despite its increase in demand and the size of the role it has in the BI goal. Thus, there is also very little research on the open source ETL tools available. The commercial ETL tools can be expensive and the ETL process must be customized to the various data sources, which may change frequently. Thus, these open-source tools have the potential to improve the process and reduce the time required to prepare the data for BI. This paper strives to compare two open source ETL python libraries and evaluate their performance to better inform what options and application the open-source tools can provide the data engineer.

# LITERATURE REVIEW

Sonali Vyas and Pragya Vaishnav did a comparative study of various ETL process and their testing techniques in data warehouse. [2] In the paper, the testing various testing methods and the difficulty of creating ETL processes.

In the paper by El-Sappagh, Hendawi, and El-Bastawissy, they proposed a model for data warehouse ETL processes. [3] The three models covered were modeling based on mapping expressions and guidelines, modeling based on conceptual constructs, and modeling based on UML environment.

Vassiliadis, Simitsis, and Skiadopoulos paper covers conceptual modeling for ETL processes. [4] The conceptual model proposed is customized for the tracing of inter attribute relationships and the respective ETL activities in the early stages of a data warehouse project.

Simitsis, Vassiliadis, and Sellis did a paper on the topic of optimizing ETL processes in data warehouses. [5] The focus of their paper is on modeling ETL processes as a state space search problem. This is a narrow focus that excludes the consideration of the physical operators and access methods. The three algorithms discussed are exhaustive search, heuristic search, and heuristic search greedy (HS greedy).

Souibgui, Atigui, Zammali, Cherfi, Yahia paper on data quality in ETL process: A preliminary study. [6] This paper covers three approaches to data quality: syntactic oriented, semantic oriented, and context oriented. The paper focused on four well-known commercial suites: Talend, Pentaho, Microsoft SQL Server, Informatica. They compared de-duplication, reliability, and performance across the four.

# METHODOLOGY

Each algorithm will consist of three sections: extract, transform, and load. In addition to the phases involved in ETL, the algorithms have a function for logging the timing of each phase. One algorithm uses the Pandas library and the second algorithm uses the Bonobo library. These differ from each other in that Pandas is a standard and very common python library used for ETL algorithms. However, the Pandas' library is not built specifically for ETL processes. The Bonobo library can be described as a fully-contained lightweight ETL framework. Bonobo seeks to utilize more of the python language's native functionality. Further, the novel Bonobo library is build specifically for the process of ETL.

# IMPLEMENTATION

The algorithms are programmed in Python and ran within the Visual Studio Code IDE. The datasets are in an online repository and extracted from a weblink. Three file types are extracted from the web. Those file types are XML, JSON, CSV of which there are three of each type. The data is moved to a folder on the computer for the extract-transform-load phases. The Pandas or Bonobo libraries are used for the extract, transform, and load functions in the python code. Once the data has been extracted and combined, then the data is transformed. The transformed data is saved to a CSV file. Some accessory python libraries are used in the program. Those libraries are glob, xml, json, zipfile, requests, and os. All the accessory libraries are used in retrieving the data from the web and unpacking the zip file so that the extract-transform-load functions can get the data from the files, manipulate the data, and then load the data to it’s save file.

# EXPERIMENTATION SETUP

**5.1 Data Sources**

For initial test runs of each algorithm, a small dataset with only two transformations is used. The dataset is contained in a zip file located in an Amazon S3 bucket owned by the IBM Developer Skills Network.[7] This data contains an individual's name with their weight and height in US units and need to be transformed to metric units.

**5.2 Evaluation metrics**

The two algorithms will be compared by the results of the following four tests.

*5.2.1 Source-to-target Count Testing*

In data, integrity is a necessity to ensure that no data is lost in the ETL process. Thus, a count of the number of source data that makes it to the target can confirm if all the data was successfully processed. This does not validate the data, only confirms it was processed.

*5.2.2. Source-to-target Data Testing*

Source-to-target data testing is used for validating that the data was transformed and load without altering the accuracy of the data. Meaning, the value of the data in the source is compared to the target data value to verify they are interrelated. This process is important if the data is medical or financial. Due to the significant time consumption of this test, the test may not be performed if it is not entirely necessary.

*5.2.3. Process Time Testing*

Each stage of the ETL process is timed. Each stage varies in the time it takes by the number of actions it must perform. Extraction depends on if the metadata file plus flat files for each data source is required (or just the flat files), and how many of each is extracted. The transformation stage varies by how many transformations are performed on each source, how many times each transformation is to be performed, and the type of transformation that is to be performed. The process time test does not time each transformation separately, just the time from start to finish for all the data and transformation to be completed.

*5.2.4. Throughput*

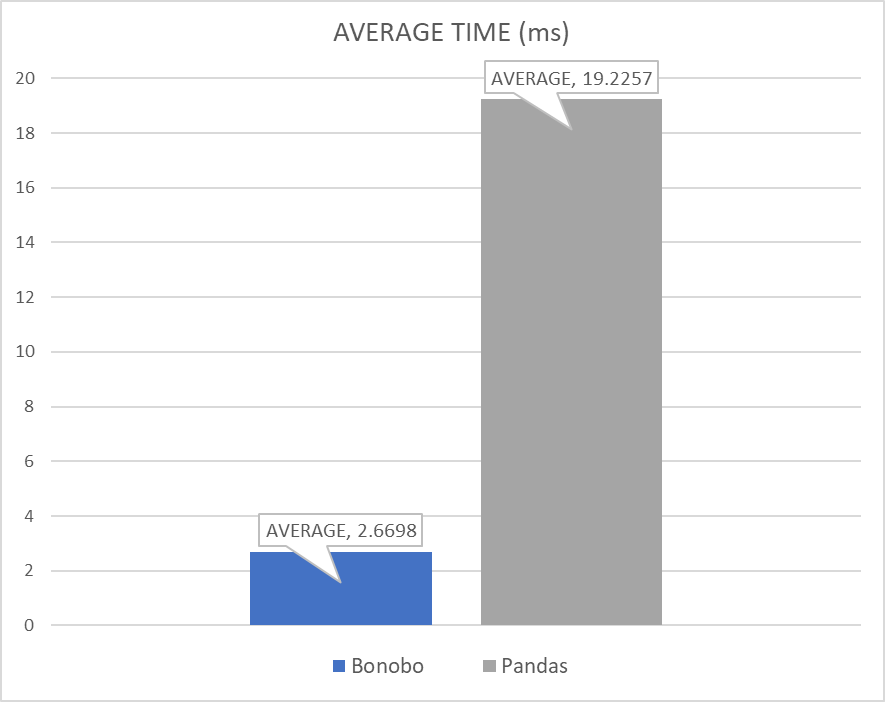
Throughput is calculated by dividing the extraction time and the overall ETL time by the number of data units. The result provides the amount of time it takes for one data unit to be processed.

# RESULTS ANALYSIS

With regard to Source-to-target count, both Pandas and Bonobo successfully extracted 78 units of data from the source.

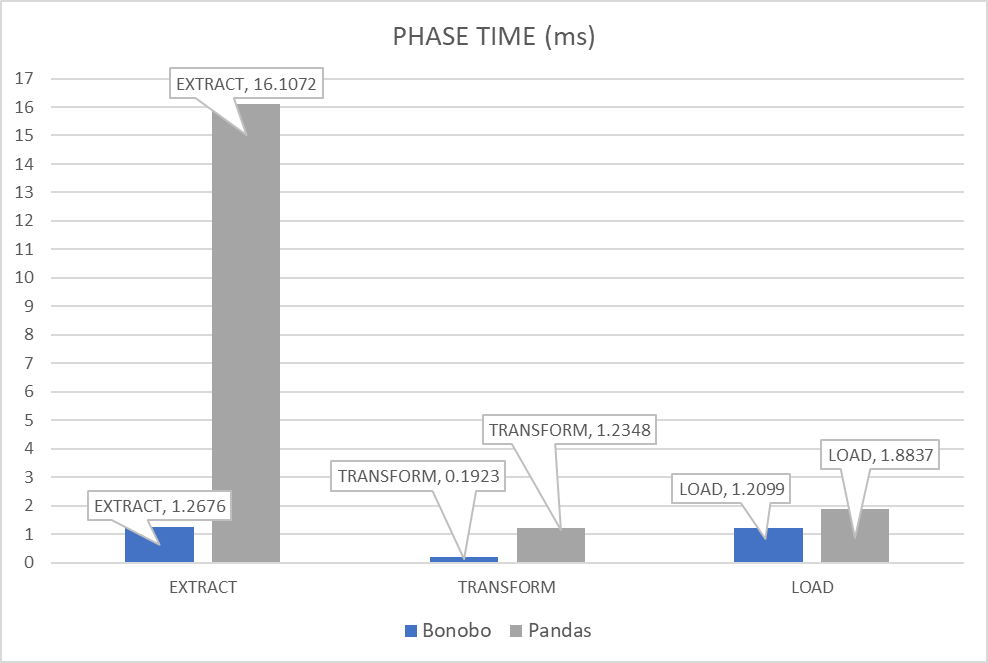
For Source-to-target data testing, both Pandas and Bonobo properly extracted the data.

For each algorithm, the processing time is measured for each part of the extract-transform-load process and the overall time for the ETL process. Pandas took an average of 19.2257 ms for the entire ETL portion of the algorithm, while Bonobo averaged 2.6698 ms.



**Figure 1. Graph of average ETL process time**

For the extract phase, Pandas took 16.1022 ms and Bonobo took 1.2676 ms. In the transform phase, Pandas took 1.2348 ms and Bonobo took 0.1923 ms. During the load phase, Pandas took 1.8837 ms and Bonobo took 1.2099 ms.



**Figure 2. Graph showing average process time by phase**

There were 78 units of data. The result is in milliseconds per unit of data.

| Throughput | Extract | Overall |
| --- | --- | --- |
| Bonobo | 0.1408 | 0.0342 |
| Pandas | 1.7897 | 1.0681 |

**Table 1. Throughput for extraction and overall**

# CONCLUSION

The two python libraries, Pandas and Bonobo, were evaluated based on their overall performance in the task of ETL and for each phase within ETL. Pandas is well known and frequently used by data engineers. Bonobo is a new python package that was created specifically for data engineering.

Both algorithms are sufficient in completing these tasks. However, the time taken to perform each step of the ETL process becomes critical when it is necessary to process large amounts of data. This was a relatively small data set, considering it was only three points of data for 26 individuals. Despite the small set, the way it is spread across various files and that two units per record needs transformation is sufficient to get an average for each algorithm. As shown in Table 1, Bonobo was significantly faster in extraction throughput and overall ETL throughput. Bonobo was faster at every phase of the process, shown in figure 2.

Bonobo does not currently have a method to extract the data from XMLs. The XML extraction was removed from the Pandas algorithm to ensure the comparison was even and balanced. The Pandas extraction time was more than halved after this changed was made. Therefore, based on the findings in this research, Bonobo would be a better choice for any data engineering project with source data in CSV or JSON files. Pandas can be used to extract data from XML files, but considering how time-consuming it is, perhaps it would be better to find an alternative. Further research can be done to determine a better package for all three of these file types.

REFERENCES

1. W.H. Inmon. *Building the Data warehouse.* Wiley, New York, NY*,* USA, 3rd edition, 2002.
2. Sonali Vyas & Pragya Vaishnav (2017) A comparative study of various ETL process and their testing techniques in data warehouse, Journal of Statistics and Management Systems, 20:4, 753-763, DOI: 10.1080/09720510.2017.1395194
3. Shaker H. Ali El-Sappagh, Abdeltawab M. Ahmed Hendawi, Ali Hamed El Bastawissy, A proposed model for data warehouse ETL processes, Journal of King Saud University - Computer and Information Sciences, Volume 23, Issue 2, 2011, Pages 91-104, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2011.05.005.
4. Panos Vassiliadis, Alkis Simitsis, and Spiros Skiadopoulos. 2002. Conceptual modeling for ETL processes. In Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP (DOLAP '02). Association for Computing Machinery, New York, NY, USA, 14–21. https://doi.org/10.1145/583890.583893
5. A. Simitsis, P. Vassiliadis and T. Sellis, Optimizing ETL processes in data warehouses, 21st International Conference on Data Engineering (ICDE'05), 2005, pp. 564-575, doi: 10.1109/ICDE.2005.103.
6. Manel Souibgui, Faten Atigui, Saloua Zammali, Samira Cherfi, Sadok Ben Yahia, Data quality in ETL process: A preliminary study, Procedia Computer Science, Volume 159, 2019, Pages 676-687, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.09.223.
7. Ramesh Sannareddy and Joseph Santarcangelo. Python Project for Data Engineering. Retrieved October 19, 2022 from https://www.coursera.org/learn/python-project-for-data-engineering/home/info
8. US National Archives. Python scripts to transform partner data for upload to National Archives Catalog. Retrieved October 19, 2022 from https://github.com/usnationalarchives/partner-data-transform