Next-generation ETL Framework to address the challenges posed by Big Data

Syed Muhammad Fawad Ali

Agenda

1. Introduction
2. Motivation
3. Extendable ETL Framework
4. Conclusion
5. Q&A

1.
INTRODUCTION

Background

Growth of Data

Volume of produced, collected, and stockpiled digital data has been continuously growing exponentially.

Expected useful data size by 2020 - 16 Trilltion GB

Dealing with Big Data

Gain great benefits in science and business.

Requires a great scientific contribution to deal with it.

Challenges with Big Data

Acquisition

Processing

Loading

Analyzing

Focus - ETL aspect of big data

TRADITIONAL ETL FRAMEWORKS

Designed for creating traditional Data Warehouse (DW), to efficiently support lightweight computations on smaller data sets

BIG DATA REQUIREMENTS

Big Data demands new and advanced computations e.g., from data cleansing or data visualization aspects.

2.
Motivation

Study on existing ETL frameworks

Focus of Study

we carried out an intensive study [1] on the existing methods for designing, implementing, and optimizing of ETL workflows.

We analyzed several techniques w.r.t their pros, cons, and challenges in the context of metrics such as:

- Support for variety of data
- support for quality metrics
- support for ETL activities as user-defined functions
- autonomous behavior

[1] S. M. F. Ali and R. Wrembel. From conceptual design to performance optimization of ETL workflows: current state of research and open problems. The VLDB Journal, pages 1–25, 2017.

Summary of the Study

Variety of Data

Limited support for semi structured and unstructured data.

Exponential growth of the variety of data format especially the unstructured and raw data

need to extend the support for processing an unstructured data along with other data formats (e.g., video, audio, binary).

Efficient Execution of WFs

Regardless of today's big ata needs - lack of emphasis on the issues of efficient, reliable, and improved execution of an ETL workflow.

No support for **user-defined functions**.

*Required techniques based on task parallelism, data parallelism, and a combination of both for traditional ETL operators as well as **user-defined functions**.*

Monitoring & Recommendation

Extensive input is required from ETL developers during the design and implementation phase of DWH and ETL development life cycle.

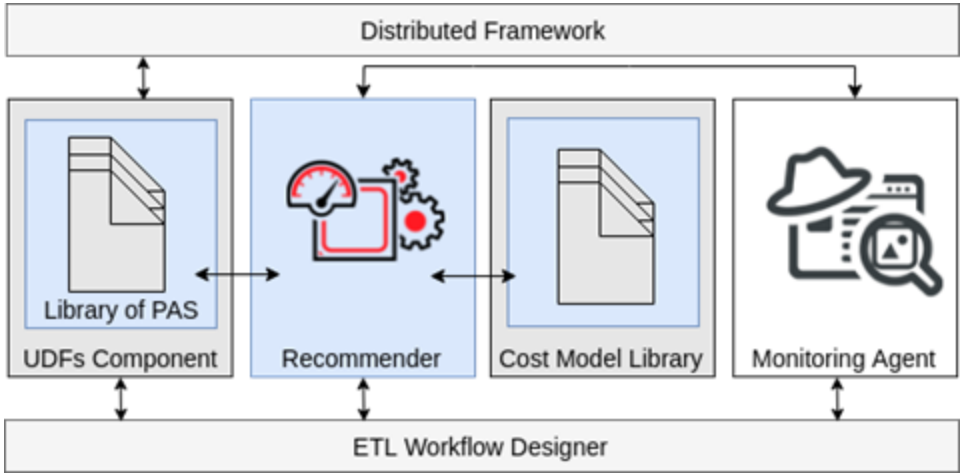
It can be error prone, time consuming, and inefficient.

Need of an ETL framework to provide recommendations on:
(1) an efficient ETL workflow design
(2) how and when to improve the performance of an ETL workflow without conceding other quality metrics

The consequence of the aforementioned observation is that designing and optimizing ETL workflows for Big Data is much more difficult than for traditional data and is much needed at this point in time.

3.
The Extendable ETL Framework

Architecture of the ETL Workflow

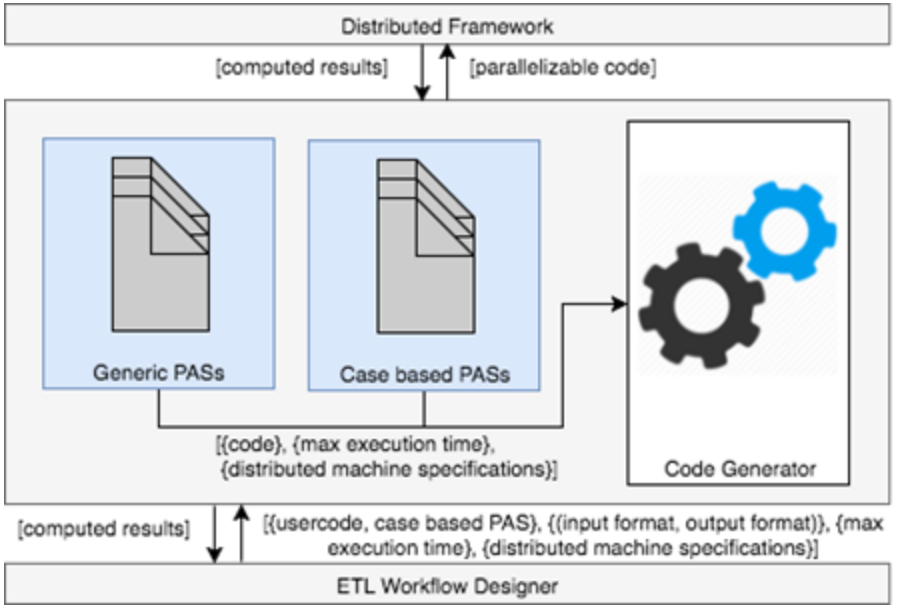


A three layered architecture:

1. **Bottom layer - WF designer**
2. **Top Layer - a distributed framework**
3. **Middle layer**
   - UDF Component
   - Recommender
   - Library - Cost Model
   - Monitoring Agent

A UDF's Component

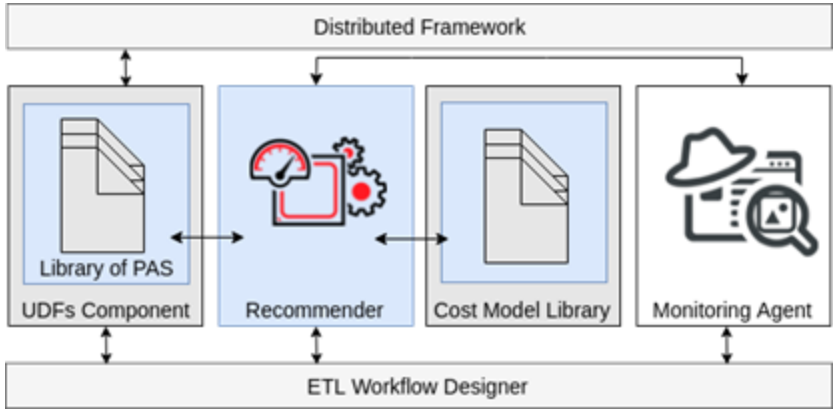
The idea behind introducing a UDFs component is to assist the ETL developer in writing a parallelizable UDF by separating parallelization concerns from the code.



A Recommender

A Recommender includes

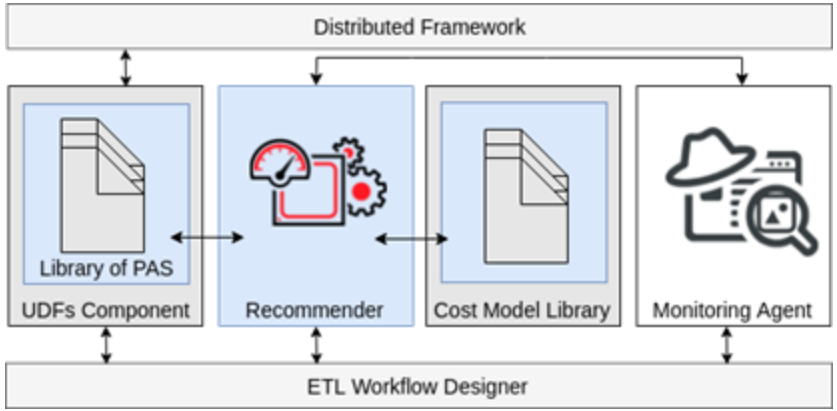
- ***an extendable set of machine learning algorithms to optimize a given ETL workflow (based on metadata collected during past ETL executions).***
- ***Metadata may be collected with the help of Monitoring Agent.***
- ***An ETL developer may be able to experiment with alternative algorithms to optimize ETL workflows (e.g., Dependency Graph approach), Scheduling Strategies)***



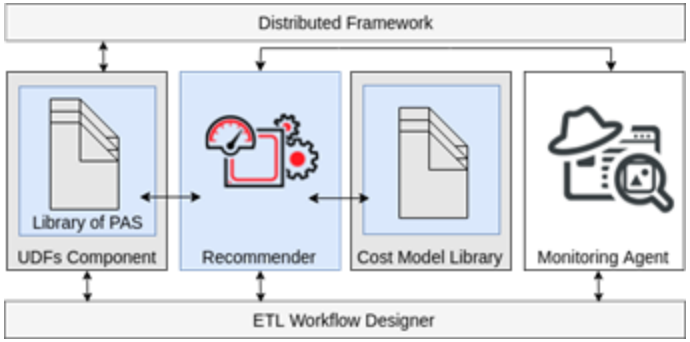
A Cost Model

- *The library of cost models may include models for:*
  - *monetary cost,*
  - *performance cost, and*
  - *both cost and execution performance*

- *A Recommender may choose the appropriate cost model from a library of cost models to make optimal decisions based on the ETL developer's input and Monitoring agent.*



A Monitoring Agent

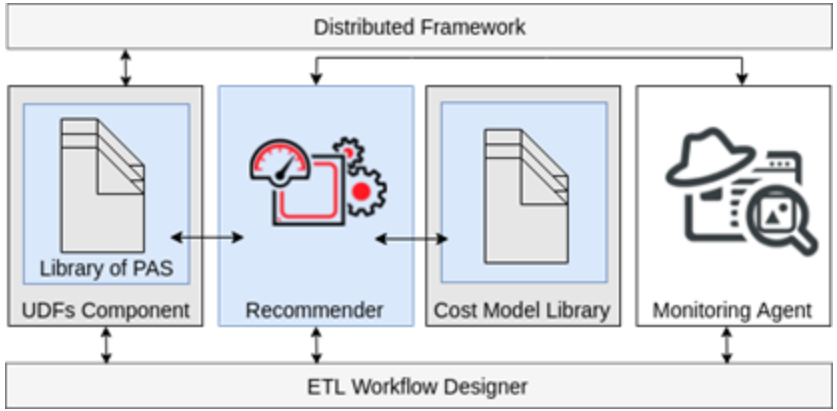


Monitoring Agent allows to:

- ***monitor ETL workflow executions***
  - *# input rows, # output rows, execution time of each step, number of rows processed per second*
- ***report errors***
  - *task or workflow failures and the possible reasons*
- ***schedule executions.***
  - *execution time of ETL workflows and creating a dependency chart for ETL tasks and workflows*
- ***gather various performance statistics.***
  - *execution time of each ETL activity w.r.t rows processed per second, execution time of the entire ETL workflow w.r.t rows processed per second, memory consumption by each ETL activity*

A Monitoring Agent

- ***Information collected by the Monitoring agent is stored in an ETL framework repository to be utilized by Recommender and Cost Model to make recommendations to the ETL developer and to generate optimal ETL workflows.***



4.
Conclusion

We believe that the proposed ETL framework is a step forward towards a fully automated ETL framework to help the ETL developers optimize ETL tasks and an overall ETL workflow for Big Data with the help of recommendations, monitoring WFs, and UDFs provided by the tool.

Currently we are working on the first steps towards building a complete ETL Framework

- ***A UDFs Component*** *- to provide the library of reusable parallel algorithmic skeletons for the ETL developer and*
- ***A Cost Model*** *- to generate the most efficient execution plan for an ETL workflow.*

Thanks!

Any questions?

You can contact me at:
fawadali.ali@gmail.com

References

1. S. M. F. Ali and R. Wrembel. From conceptual design to performance optimization of ETL w orkflows: current state of research and open problems. The VLDB Journal, pages 1–25, 2017.
2. S. K. Bansal. Tow ards a semantic extract-transform-load (ETL) framework for big data integration. In Proceedings of International Congress on Big Data, pages 522–529. IEEE, 2014.
3. J. Duggan, A. J. Elmore, M. Stonebraker, M. Balazinska, B. How e, J. Kepner, S. Madden, D. Maier, T. Mattson, and S. Zdonik. The BigDAWG Polystore System. SIGMOD Record, pages 11–16, 2015.
4. T. Ibaraki, T. Hasegaw a, K. Teranaka, and J. Iw ase. The multiple choice knapsack problem. Journal of Operations Research Society Japan, pages 59–94, 1978.
5. A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. Performance analysis of cloud computing services for many-tasks scientific computing. Transactions on Parallel and Distributed systems, pages 931–945, 2011.
6. K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright. Performance analysis of high performance computing applications on the amazon w eb services cloud. In International Conference on Cloud Computing Technology and Science, pages 159–168. IEEE, 2010.
7. A. Karagiannis, P. Vassiliadis, and A. Simitsis. Scheduling strategies for efficient ETL execution. Information Systems, pages 927–945, 2013.
8. M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob. Big IoT data analytics: Architecture, opportunities, and open research challenges. IEEE Access, pages 5247–5261, 2017.
9. B. Martinho and M. Y. Santos. An architecture for data warehousing in big data environments. In Proceedings of Research and Practical Issues of Enterprise Information Systems, pages 237–250. Springer, 2016.
10. A. Simitsis, P. Vassiliadis, and T. Sellis. State-space optimization of ETL w orkflows. IEEE Transactions on Know ledge and Data Engineering (TKDE), pages 1404–1419, 2005.
11. A. Simitsis, K. Wilkinson, U. Dayal, and M. Castellanos. Optimizing ETL w orkflows for fault-tolerance. In Proceedings of IEEE International Conference on Data Engineering (ICDE), 2010.
12. I. Terrizzano, P. Schw arz, M. Roth, and J. E. Colino. Data Wrangling: The Challenging Journey from the Wild to the Lake. In Proceedings of Conference on Innovative Data Systems Research (CIDR), 2015.
13. V. Viana, D. De Oliveira, and M. Mattoso. Tow ards a cost model for scheduling scientific workflows activities in cloud environments. In Proceedings of IEEE World Congress on Services, pages 216–219, 2011.