Metamorfose: a data transformation framework based on Apache Spark

Evandro M. Kuszera *Leticia M. Peres* *Marcos Didonet Del Fabro*

UTFPR-Dois Vizinhos *UFPR - C3SL Labs* *UFPR - C3SL Labs*

evandrokuszera@utfpr.edu.br, {lmperes,marcos.ddf}@inf.ufpr.br

Abstract:

On the context of large availability of open data, it is important to provide interactive solutions where a data transformation workflow can be easily deployed and developed. In this paper we will present the Metamorfose tool, a framework for data transformation built on large-scale data processing engine called Apache Spark[1]. Through a graphical user interface is possible to define an interactive data transformation workflow where the user can loading data, defining mappings, performing transformations, exploring and persisting the results. A mapping definition can be integrated with data transformation functions implemented in Java or Javascript, providing a flexible way to define complex transformations. It is possible to execute SQL queries to filter, aggregate and join the datasets before or after every data transformation. The results can be persisted as CSV file or relational table.

Metamorfose:

Through the graphical user interface it is possible to load data source for transformation, execute data queries through Spark SQL (Apache Spark module) and specify mappings[2] between source and target schema. Each data source is loaded as a Spark dataset. To transform the data, map functions are executed on the source dataset according to the user-defined mappings. In Figure 1 we can see Metamorfose[4,5] architecture.

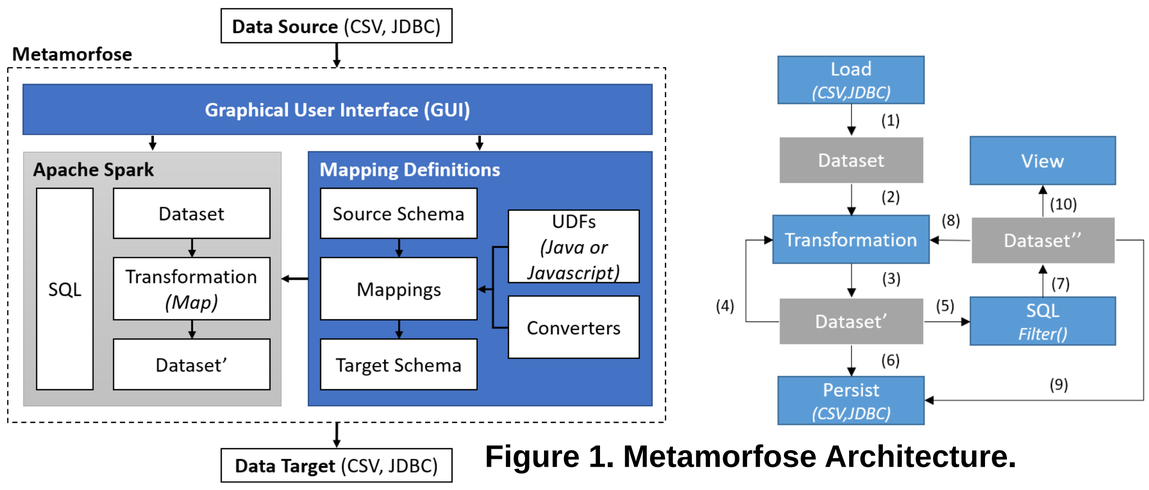

Figure 1. Metamorfose Architecture.

Transformation Example:

Figure 2 (a) shows an example of transformation over a table with person data. ID field is mapped to the COD field in the target schema. NAME and LASTNAME fields are mapped to the NAME field. SEX field data are transformed from 'F' and 'M' to numerical values 1 and 2, respectively. ADDRESS and PHONE fields only exist in the target schema and they will receive an empty string and null value, respectively. Figure 2 (b) shows the mappings provided by the user.

ID	NAME	LASTNAME	SEX
1	João	Silva	M
2	Maria	Silva	F
3	José	Silva	M

⇩

COD	NAME	SEX	ADDRESS	PHONE
1	João Silva	2	""	null
2	Maria Silva	1	""	null
3	José Silva	2	""	null

Figure 2(a). Source and target entities.

#	SOURCE	TARGET	TYPE	SCRIPT
1	ID	COD	Casting	
2	$LIST(NAME, LASTNAME)	NAME	Function	function concat (value) { return value[0] + ' ' +value[1]; }
3	SEX	SEX	Function	function sex (value) { if (value[0]=='F') return 1; else if(value[0]=='M') return 2; }
4	$VALUE('')	ADDRESS	Casting	
5	$VALUE(NULL)	PHONE	Casting	

Figure 2(b). List of mappings.

Data Transformation with Metamorfose:

The example above can be executed with Metamorfose through three steps: loading data, mapping fields and executing transformations. The Figures 3, 4 and 5 show these three steps.

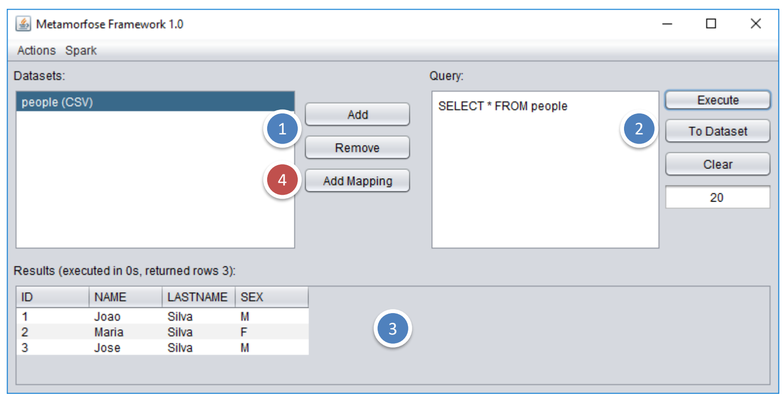

Figure 3. Main screen.
1 – Loading dataset.
2 – Querying dataset.
3 – Visualizing dataset.
4 – Adding mappings to dataset.

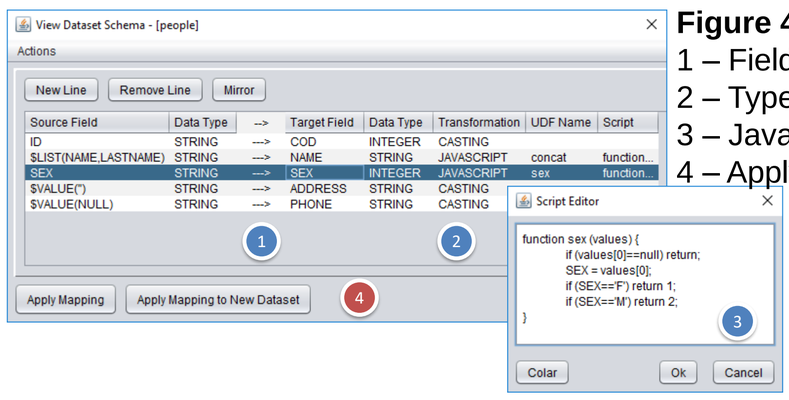

Figure 4. Mapping screen.
1 – Field mappings.
2 – Type of transformation and UDF.
3 – JavaScript UDF.
4 – Applying mappings to dataset.

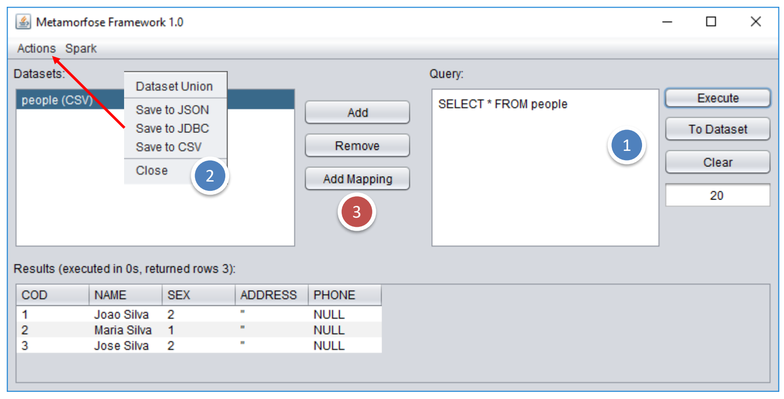

Figure 5. Transformed dataset.
1 – Executing queries.
2 – Persisting dataset.
3 – Using it in another data transformation.

Experiments:

We have performed experiments using Brazilian enrollment data of the year 2013 provided by INEP[3]. Metamorfose was used to map and transform CSV data to a predefined PostgreSQL database. It were defined 82 mapping fields between CSV file and target relational schema. One-to-one, many-to-one mappings and Javascript UDFs were used to transform around 55 million records. Metamorfose was run on a machine with Intel Core i7 2.5GHz processor, 16GB RAM, Windows 10 Home and Postgres 10. The Table 1 shows the number of records and execution time.

Brazil Region	Records	Time
Midwest	4.038.979	10 min
South	7.276.108	18 min
Northeast	16.729.543	41 min
Southeast + North	27.379.690	65 min

Table 1. Number of records and execution time.

Referências:

1. Zaharia et al. (2016). *Apache spark: A unified engine for big data processing*. Commun. ACM, 59(11):56–65.
2. Dessloch, S., Hernandez, M. A., Wisnesky, R., Radwan, A., and Zhou, J. (2008). Orchid:Integrating schema mapping and etl. In 2008 IEEE 24th ICDE, pages 1307–1316.
3. INEP: http://www.inep.gov.br/
4. Demo video: https://youtu.be/ta9mXuCeIwM
5. Metamorfose repository: https://github.com/evandrokuszera/metamorfose

UFPR UNIVERSIDADE FEDERAL DO PARANÁ

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ CAMPUS DOIS VIZINHOS

C3SL Centro de Computação Científica e Software Livre

SBC