Data Wrangling at Scale

The experience of EW-Shopp

Nikolay Nikolov
SINTEF
Oslo, Norway
nikolay.nikolov@sintef.no

Michele Ciavotta
University of Milan-Bicocca
Milan, Italy
michele.ciavotta@unimib.it

Flavio De Paoli
University of Milan-Bicocca
Milan, Italy
flavio.depaoli@unimib.it

ABSTRACT

This paper presents a subsystem of a comprehensive platform dedicated to data transformation, linking and extension of large data sets. Furthermore, we detail and discuss both the main requirements that have led to the design and development of the platform, and the devised approach, which is a direct outcome of the requirement elicitation and discussion phase. In particular, the platform supports both design and run time aspects of the data transformation process, which is reflected in the architecture. Some initial tests have been carried out on a prototype implementation of our architecture on data sets of ~1TB featuring promising performance.

KEYWORDS

Big Data Processing, Data Wrangling, Data Integration, Data Enrichment, Data Extension, Linked Data

ACM Reference Format:
Nikolay Nikolov, Michele Ciavotta, and Flavio De Paoli. 2018. Data Wrangling at Scale: The experience of EW-Shopp. In *Proceedings of XXXX (XXX).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the past years, Big Data has been receiving ever more attention both in academia and industry. This can be partly explained by the promise of Big Data advocates to create value (estimates indicate yearly earnings in the order of 200 billion dollars by 2020 [2]) delivering more accurate and efficient data-driven decision-making processes via analytics. The term *Data Analytics*, however, is vague and too often used to define a class of heterogeneous activities such as *data wrangling* [1] (that includes preparation processes as cleaning, linking, enrichment/extension), training and application of analytic models, up to business intelligence and visualization. In this paper, we focus on data wrangling on tabular data, as the correct definition of data preparation processes is a painful and time-consuming (up to the 80% of total time [3] for performing analytics) hurdle in the implementation of value-added data-driven pipelines. In particular, we describe a software solution (part of a larger ecosystem) tailored to perform data wrangling and generation of linked data, which is able to operate on massive data sets with a special support provided to the integration of weather and events data. This platform has been designed and realized following the guidelines of the Lean methodology, based on a rigorous activity of requirements elicitation that took into account both literature and best practices as well as the needs expressed by involved stakeholders.

Although there are many solutions for processing Big Data, most available platforms feature a general-purpose processing model designed for users familiar with programming languages and process definition, but usually inexperienced in the particular domain to which the data pertain [8]. For this reason, they do not provide specific and user-friendly tools for the definition of data transformation pipelines [5] (design time), and, instead, often focus only on the management of their execution (run time). The main implication is the emergence of a two-pronged working environment consisting, on the one hand, of domain experts in charge of designing data transformations and, on the other hand, of engineers who deploy them into a production environment. This socio-occupational gap between interdependent groups pigeonholed in strictly separated roles can cause issues and delays in the development and maintenance of Big Data solutions and calls for specific solutions to be bridged.

The proposed platform for designing and executing data transformation pipelines (from the cleaning phase to linking and publication) is being realized in the context of EW-Shopp. This is a H2020 EU project aiming at supporting the e-commerce, retail, and marketing industries in improving their efficiency and competitiveness through enabling them to perform predictive and prescriptive analytics over integrated, enriched, and extended large data sets using open and flexible solutions. This paper describes the architecture of a subsystem of the EW-Shopp[1] integrated platform aimed at facilitating and enhancing data transformation, integration, and extension processes with the focus on the large-scale integration of weather and event information.

The paper is structured as follows. In Section 2 the main design principles that have driven the definition of the architecture are presented whereas the components and workflow are discussed in Section 3. Finally, Section 4 concludes the document.

2 PLATFORM DESIGN

In this section, we outline the guiding principles behind the current platform architecture. In order to precisely identify the main application scenarios and objectives, a thorough requirement gathering phase has been carried out taking into account the first principles underpinning Big Data solutions and the expectations of the business partners involved in the project. The main requirements drove the core design of the platform. As the platform deals with issues of

[1] http://www.ew-shopp.eu/

XXX, XXXX, XXXX

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

data processing, integration, enrichment, and extension, it needed to support data flow definition and execution. Furthermore, EW-Shopp business cases provided input data sets in tabular format from heterogeneous domains, and of different volumes, including Big Data. In many cases, the data provided by organizations was a core asset and thus, also data confidentiality needed to be retained. Due to the varying security requirements of organizations, the system had to be designed to flexibly support different security setups. The data flows themselves had to be defined on a high level and composed of independent steps, which also include data integration and enrichment at scale. Finally, deployment of the flows had to be flexible in terms of both hardware and software, so that they can be deployed in different infrastructures that fit organizational needs best. The main specific platforms for data wrangling available on the market have been also evaluated, and in particular, OpenRefine[2], Trifacta Wrangler[3] and Karma[4].

At the end of this process, the need for an *open source* platform, capable of managing data in *tabular format* and of generating *linked data*, became clear. In addition, two possible use cases have emerged. In the first case, small amounts of data have to be manipulated, the platform has to be lean and *easy to install on a commodity machine* similarly to tools like OpenRefine. In the second scenario, a genuinely big data must be managed, this means that a domain expert user must be able to describe the transformation through a user-friendly and interactive interface and to execute it in an automated way as a *batch* process (as in Karma and Trifacta Wrangler). Common to both scenarios is the need to supply linking and enrichment services to manage *geocoding*, *schema linking*, *entity linking* and extension with *weather* and *events* data.

In an attempt to give a syncretic response to the needs outlined in the requirements, we have based platform design on two fundamental driving principles: *Separation of Concerns* and *Dimension Reduction*. In particular, by applying separation of concerns, we have conceived the platform as composed of three tiers (Figure 1):

Core Data Services (in light blue): These components provide access to corporate on-boarding or third party data. These data sets are used in both data linking and extension processes. Figure 1 shows services to access the project's core data, i.e., weather (W), events (E) and products (P). Other enrichment sources may include services as Wikifier[5], and freely accessible knowledge bases such as DBpedia[6].

Platform Services (in green): data preparation, analytics and visualization services (the last two are not addressed in this paper). These services offer simple and intuitive user interfaces for creating (and executing if the working table is small enough) data wrangling, linking and extension pipelines.

Corporate Services (in red): These services implement platform components needed for data governance, i.e., ingestion, storage, processing, data flow and security management of massive data sets. This tier is used to perform data wrangling operations at scale.

[2]http://openrefine.org/
[3]https://www.trifacta.com/products/wrangler/
[4]http://usc-isi-i2.github.io/karma/
[5]http://wikifier.org/
[6]https://wiki.dbpedia.org/

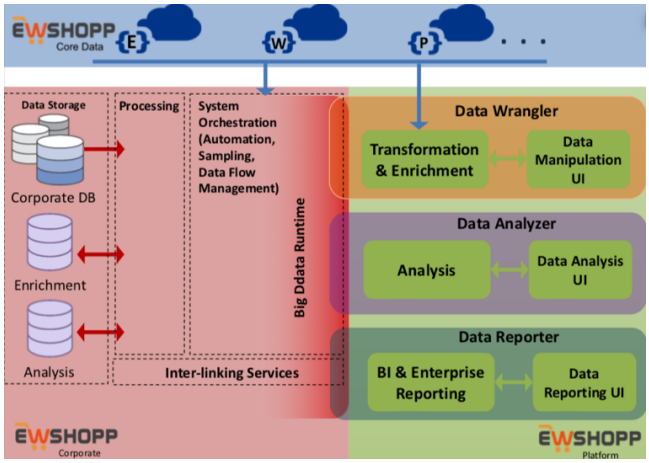

Figure 1: General architecture

The rationale of the Dimension Reduction [6, 9] approach in Big Data is also rather simple in its general lines; it consists in reducing the size of the data set by identifying a possible compact representation (i.e., a subset of data plus a profile). In this way, the data transformation operations can be designed and tested on a set that can be several orders of magnitude smaller than the original. This should ensure greater responsiveness and efficiency for the applications involved without negatively affecting accuracy. For this reason, we have introduced into the platform a specialized component, whose task is to properly generate this data subset. We remark that the presented design choices do not reduce the applicability and generality of the presented architecture as, where it is possible (e.g., in cases where the data to be transformed is manageable), the corporate subsystem can be excluded. In this mode, the user can work directly against the original data set.

3 DATA WRANGLING AT SCALE

As mentioned above, in the designed architecture a logical separation between platform and corporate services has been imposed. This structure is conceived to make the architecture flexible and adaptable to different deployment scenarios. In what follows, the main components and interaction of the platform are presented.

Figure 2 shows a component diagram of our wrangling solution with component interactions and information flows. The image presents the most general case, where the data set to be transformed is genuine Big Data and, for this reason, it is managed by the corporate services. In this scenario, components from all logical areas are involved; we have on the left a corporate component called *Sampler* , the *Summarizer* and the *Engine* that are the components that are entrusted of creating the (reduced) working data set to be used to define the transformations, of providing a set of suggestions for the table annotation process based on summaries of existing knowledge bases or previous annotations, and finally, of interacting with the *Big Data Runtime* to carry out on a large scale the transformations defined by the user, respectively. The definition of the transformation pipeline, instead, is carried out using the platform services, depicted on the right side of the Figure.

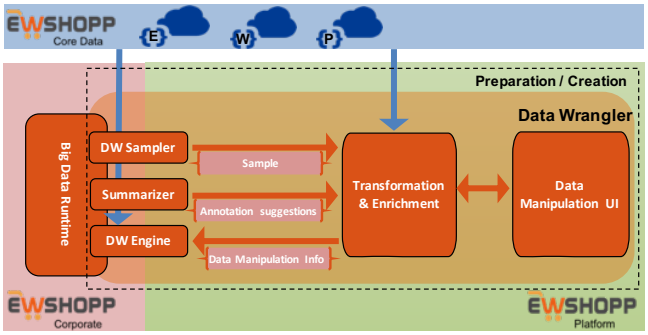

Figure 2: Data Wrangler's components and interactions

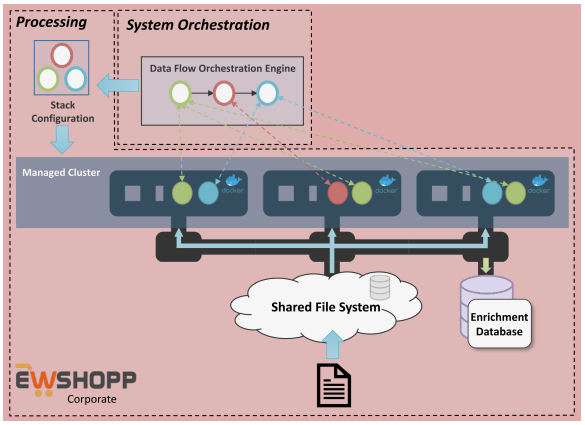

Figure 3: Big Data Runtime components interaction

In particular, the user is supported in this task by a graphical manipulation interface, which in turn interacts with a *Transformation and Enrichment* back-end service. This service has the twofold duty of executing the pipeline on the reduced data set, thus allowing the platform to be used as a standalone solution, and of interacting with the corporate services. In both cases, core data services support linking and extension capabilities.

The architecture and components interactions within the Big Data Runtime are depicted in Figure 3. The *System Orchestration* sub-component is used to define the high-level data flows that will be executed by the *Processing* component. Such Data flows include the data wrangling pipelines but they may also incorporate pre- and post-processing steps, e.g, automatic obtaining of data, re-formatting, import to a data warehouse (enrichment database). The end result is a data flow that can be deployed as a Function as a Service (FaaS) computing service over a managed cluster of resources, which allows for easy integration with business processes and heterogeneous infrastructures (see Section 3.2).

For sake of clarity a high-level workflow is reported:

(1) A reduced data set (possibly complemented by profiling information) is created and passed to the *Transformation and Enrichment* component.
(2) The user operates the *Data Manipulation UI*, which also interacts with the Core Data services and *Summarizer* to perform schema and entity linking, to extend the data with weather and events information.
(3) The application generates a self-contained machine-runnable pipeline of the user's operations which is eventually executed on the original data set by the *Big Data Runtime*.

The next two subsections detail the components and interactions of the proposed solution, highlighting which components are involved at design time and which at run time.

3.1 Platform services

The Data Wrangler is a composite component built upon the DataGraft platform [7], extended with semantic enrichment and Big Data processing capabilities. DataGraft provides tools for cleaning and transformation of tabular data into RDF and graph generation mappings. DataGraft comes with an interactive web application, named Grafterizer [8], which serves as a graphical interface helping platform users to transform data from a tabular format into a graph format. The transformation supports both cleaning and graph mapping steps. Transformation steps on rows (add, drop, filter, duplicate detection etc.), columns (add, drop, rename, merge etc.) and entire data set (sort, aggregate etc.) are provided together with visualization of the result after each step. The mapping to ontologies or vocabularies is performed on the cleaned-up data.

As mentioned, the EW-Shopp platform aims to support scalable data enrichment. Therefore, DataGraft and the transformation tool, Grafterizer, incorporate two sub-components called ASIA and ABSTAT. The components provide functionalities for the semantic enrichment of data tables and profiling of knowledge graphs, respectively. ASIA is meant to aid users in integrating business data and can be used to map the data schema to shared vocabularies/ontologies, or link data values to shared systems of identifiers, which enables the extraction of additional data from third-party sources and their fusion into the original tabular data. Schema-level and instance-level links are created by ASIA as annotations for the table. ABSTAT [4] is a tool to profile knowledge graphs represented in RDF, based on linked data summarization mechanisms. The profiles extracted by ABSTAT describe the content of knowledge graphs, using abstraction (schema-level patterns) and statistics. Such profiles are exploited by ASIA to provide the user with suggestions in schema linking activity.

3.2 Corporate Services

The Corporate services of the EW-Shopp solution are mainly responsible for the orchestration and execution of wrangling operations at scale. In particular, based on the user-defined data wrangling pipeline (cleaning, transformation, graph mapping, enrichment, and extension) declared on the sampled data, a so-called transformation model is created and transmitted to the corporate services in order to be applied to the full data set. The transformation model is generated by the Data Wrangler component and packaged (compiled) in a self-contained executable Java archive (JAR) directly from the user interface of Grafterizer. The self-contained executables are compiled at runtime by using the fact that the Grafterizer data wrangling pipelines are translated to Clojure[7] code, which can be

[7] https://clojure.org/

executed on the Java Virtual Machine. This JAR is then used as input to the main step of a larger process that is referred to as Data Flow. Data Flows are an extension of the data wrangling pipeline with pre- and post-processing actions. To serve as an example, a data flow coming from a business case of EW-Shopp is described below:

(1) Decompress data - large data set is delivered as a set of archives contain CSV files, which are stored in a shared file system of our private cloud.
(2) Split data in chunks for faster processing and horizontal scalability of inputs.
(3) Execute the data wrangling pipeline - Clean up, re-format, transform, map to ontology, and enrich the split data.
(4) Import the resulting data set into the Enrichment database

The System Orchestration component provides a high-level interface to handle and monitor data flows, whereas the Processing component is in charge of carrying them out. From the technological point of view, data flows are compiled into a chain of Docker[8] containers that are in turn deployed and run through a Container Orchestration system (e.g., Kubernetes[9], Rancher[10]) on a cluster of machines connected via an Ethernet fabric and mounting a shared filesystem (e.g., GlusterFS[11]). The implementation of a container-based solution has several benefits; for instance, it allows to decouple the data flow deployment from the particular stakeholder's hardware infrastructure also working in heterogeneous distributed environments. Furthermore, it guarantees flexible deployments, better resource utilization, and seamless horizontal scalability.

Finally, for the sake of completeness, the following non-secondary aspects of the execution of a data flow are presented:

Deployment of the flow configuration - the flow configuration contains the context information to be fed to each step that composes the data flow, as well as any additional hardware constraints for each step. These parameters are passed to the containers in the chain they are intended for. Notice that an extensible library implements the most common (parametrized) actions to be used in the definition of a data flow. Moreover, each action implements a common interface to guarantee both composability and observability.

Execution of the data flow - The Processing component establishes a suitable deployment in terms of machines and resources to be used and starts the appropriate number of containers for each step of the flow. Each of the steps consists of a set of containers that work independently and in parallel and can be scaled up or down on demand if required. The communication between two consecutive steps of the chain, that is, the handover of the partial results, occur through writing and reading from a file system in a specific way, defined in the context information.

Data Storage - The results of the data flow can be stored on disk or (more typically) imported to an Enrichment Database. The database serves as a single source of truth to be used, for example, to slice and dice the data to obtain a subset needed for analytics or machine learning jobs.

[8]https://www.docker.com/
[9]https://kubernetes.io/
[10]https://rancher.com/
[11]https://www.gluster.org/

4 CONCLUSIONS

In this paper, we have outlined the architecture of a platform designed to transform, link and extend massive data sets. The architecture features three logical tiers with distinct tasks, providing support for both design-time and run-time (definition of data clean-up and transformation, definition of data flows and their execution). As of the time of the writing of this paper, the EW-Shopp platform is under active development. A prototype of the platform has been deployed and tested for project business cases. Moreover, preliminary experiments have been carried out where the prototype has been used to successfully execute a data flow to continuously clean-up, transform, enrich sample data in the magnitude of ~1TB data demonstrating the viability of the approach and showing promising performance.

ACKNOWLEDGMENTS

The work in this paper is partly supported by the H2020 project EW-Shopp (Grant number: 732590).

REFERENCES

[1] Tim Furche, Georg Gottlob, Leonid Libkin, Giorgio Orsi, and Norman W Paton. 2016. Data Wrangling for Big Data: Challenges and Opportunities.. In *EDBT*. 473–478.
[2] IDC. 2017. Worldwide Semiannual Big Data and Analytics Spending Guide. https://www.idc.com/getdoc.jsp?containerId=prUS42371417
[3] Steve Lohr. 2014. For big-data scientists,'janitor work'is key hurdle to insights. *New York Times* 17 (2014).
[4] Matteo Palmonari, Anisa Rula, Riccardo Porrini, Andrea Maurino, Blerina Spahiu, and Vincenzo Ferme. 2015. ABSTAT: linked data summaries with abstraction and statistics. In *International Semantic Web Conference*. Springer, 128–132.
[5] Erhard Rahm and Hong Hai Do. 2000. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.* 23, 4 (2000), 3–13.
[6] Julian A Ramos Rojas, Mary Beth Kery, Stephanie Rosenthal, and Anind Dey. 2017. Sampling techniques to improve big data exploration. In *2017 IEEE 7th Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE, 26–35.
[7] Dumitru Roman, Nikolay Nikolov, Antoine Putlier, Dina Sukhobok, Brian Elvesæter, Arne Berre, Xianglin Ye, Marin Dimitrov, Alex Simov, Momchill Zarev, et al. 2016. DataGraft: One-stop-shop for open data management 1. *Semantic Web* Preprint (2016), 1–19.
[8] Dina Sukhobok, Nikolay Nikolov, Antoine Pultier, Xianglin Ye, Arne Berre, Rick Moynihan, Bill Roberts, Brian Elvesæter, Nivethika Mahasivam, and Dumitru Roman. 2016. Tabular data cleaning and linked data generation with Grafterizer. In *International Semantic Web Conference*. Springer, 134–139.
[9] Muhammad Habib ur Rehman, Chee Sun Liew, Assad Abbas, Prem Prakash Jayaraman, Teh Ying Wah, and Samee U Khan. 2016. Big data reduction methods: a survey. *Data Science and Engineering* 1, 4 (2016), 265–284.