Tools and data integration for eScience platforms

Romulo Goncalves[1], Niels Drost[1], Stefan Verhoeven[1], Jisk Attema[1], and Jason Maassen[1]

[1]NLeSC Amsterdam, The Netherlands {*{r.goncalves,n.drost,s.verhoeven,j.attema,j.maassen}@esciencecenter.nl*}

***Abstract*—**

I. Introduction

The tools and data integration work of Sherlock project aims to provide easy and efficient data injection into a HDFS cluster, batch processing, interactive processing and export of results to external processing or storage systems. The decision to use HDFS as a data hub is due to its utilization in Hansken, NFI system for forensic research, but also to exploit current state-of-the-art solutions for Big Data exploration such as Spark. The diagram in Figure 2 shows how each component is interlinked with a Hadoop 2.0 cluster. It has a data import docker to convert different file types to the most appropriated file format supported by Hadoop.

The choice of its format is based on the type of processing with the aim to improve processing efficiency and have a low storage footprint. To exploit data locality and fault-tolerance on a Hadoop Cluster, external libraries are dockerized and through MRdocker, it instantiates a docker image in the Map phase of a MapReduce job, or SprDocker, it instantiates a docker image on each data node, are executed. The input, the intermediate, or the final result is made accessible through the Scala, the R, and the Python interfaces of Spark for a step-wise forensic exploration. Such interfaces allow the users to develop web-apps using R-shiny, or Python fask to feed Java Script web-pages. It provides a user-friendly visualization with all the heavy computations being done by Spark. In case the user wants to process intermediate results somewhere else, a bridge through a NFS mount or through a specialized data exporter is provided by our solution. For example, it allows the data to be loaded into external database for later re-use or be consumed by an external data visualization tool running in a docker.

The remainder of the paper is as follows. Section **??** discusses the general architecture. In Section **??**, through different use case scenarios, flexibility and efficiency on exploring climate and geo-spatial data is shown. Finally the article ends with a summary in Section **??** and future plans in Section **??**.

II. Background

III. Evaluation

IV. Future work

—-Towards Generating ETL Processes for Incremental Loading— We presented an approach for generating incremental load processes for data warehouse refreshment from declarative schema mappings . The basis of our work has been provided by Orchid, a prototype system developed at IBM Almaden Research Center, that translates schema mappings into ETL processes and vice versa. Orchid-generated ETL processes, however, are limited to initial load scenarios , i.e. source data is exhaustively extracted and the warehouse is com- 109 pletely (re)built. Incremental load processes, in contrast, propagate changes from the sources to the data warehouse. This approach has clear performance benefits. Change Data Capture (CDC) and Change Data Application (CDA) tech- niques are used to capture changes at the sources and re- fresh the data warehouse, respectively. We defined a model for change data to characterize both, the output of CDC techniques and the input of CDA techniques. Since CDC techniques may su from limitations we introduced a notion of partial change data . We discussed the propagation of partial change data within incremental load processes. Our approach allows for reasoning on how limitations of CDC techniques determine the set of applicable CDA techniques. That is, it allows inferring satisticable CDA reuirements from given CDC limitations and, the other way round, acceptable CDC limitations from given CDA requirements. We further, demonstrated the exploitation of properties of data sources (such as schema constraints) to reduce the complexity of incremental load processes. By leveraging Or- chid's deployment facilities, we are able to generate exe- cutable ETL processes. We are confident that our work contributes to the improvement of ETL development

V. Summary