Development of DKB ETL module in case of data conversion

A Y Kaida[1], M V Golosova[2], M A Grigorieva[2] and M Y Gubin[1]

[1] Tomsk Polytechnic University, 30, Lenina Ave., Tomsk, 634050, Russia
[2] National Research Center "Kurchatov Institute", 1, Akademika Kurchatova Sq., Moscow, 123182, Russia

E-mail: ayk13@tpu.ru

Abstract. Modern scientific experiments involve the producing of huge volumes of data that requires new approaches in data processing and storage. These data themselves, as well as their processing and storage, are accompanied by a valuable amount of additional information, called metadata, distributed over multiple informational systems and repositories, and having a complicated, heterogeneous structure. Gathering these metadata for experiments in the field of high energy nuclear physics (HENP) is a complex issue, requiring the quest for solutions outside the box. One of the tasks is to integrate metadata from different repositories into some kind of a central storage. During the integration process, metadata taken from original source repositories go through several processing steps: metadata aggregation, transformation according to the current data model and loading it to the general storage in a standardized form. The R&D project of ATLAS experiment on LHC, Data Knowledge Base, is aimed to provide fast and easy access to significant information about LHC experiments for the scientific community. The data integration subsystem, being developed for the DKB project, can be represented as a number of particular pipelines, arranging data flow from data sources to the main DKB storage. The data transformation process, represented by a single pipeline, can be considered as a number of successive data transformation steps, where each step is implemented as an individual program module. This article outlines the specifics of program modules, used in the dataflow, and describes one of the modules developed and integrated into the data integration subsystem of DKB.

1. Introduction

Modern high energy nuclear physics (HENP) experiments, such as those at Large Hadron Collider (LHC) [1] facilities, produce huge volumes of meta information about the experiment inner processes and the project as a whole. This information is stored in different metadata storages, which exist autonomously. It makes the gathering and interlinking of HENP experiments metadata a complex issue. In some cases, when it is needed to reproduce previous experiments results, due to the autonomy of information systems, scientists have to perform search through heterogeneous data sources to get all the required information about previous researches by hand, because there are no automatic tools for this task [2].

The Data Knowledge Base R&D project [3] was initiated by ATLAS [4] experiment on LHC to solve this issue and provide fast and easy access to significant information about LHC experiments for scientific community.

2. Semantic Web usage

According to the main goal of DKB, the authors decided to choose a set of Semantic Web [5, 6] standards for presenting knowledges about scientific experiments. These standards allow one to create a plausible and flexible structure which represents all the objects in the subject field as well as their interconnections. Using the ontological approach, the authors created an open model of the domain, represented as a semantic web. As opposed to relational databases which are designed to work only with homogeneous data, knowledge bases using the Semantic Web approach are capable of storing and querying heterogeneous data and metadata [6].

A general method of information modeling in Semantic Web is Resource Description Framework (RDF). It states that any concept of information field should be described in a form of triples (subject–predicate–object statements), containing information of objects and their relationships, but does not define the exact data format to store these triples. To host the central RDF repository of DKB, the Virtuoso Universal Server was chosen because of its RDF processing capabilities.

For the automatic data integration, several dataflow pipelines were developed for different metadata sources, implementing extraction of the data from the source and applying the required analysis and transformation steps to them. These pipelines [7] essentially provide possibility to organize message-based stream processing for data coming from different sources, minimizing delay between events ‘data extracted from the source’ and ‘data loaded to the final storage’ [8, 9].

The final step, common for every pipeline from an original source to the DKB RDF storage, is to load results to the Virtuoso. But right before this stage there must be one more step (this time – individual for every pipeline) to prepare data for the upload. This step by implication is a data converter that transforms data from some internal pipeline format into the RDF one.

The metadata sources can contain different types of data: structured, semi-structured, quasi-structured and unstructured [3]. A substantial part of HENP experiments metadata belongs to the unstructured type. However, there are no automatic tools for unstructured metadata conversion, as there is no universal rule fitting all possible use cases. Thus, for every case the converter step must be developed individually, addressing the specifics of the original data and implementing set of transformation rules for this particular case. This work is dedicated to the development of this converter step for one of the DKB dataflow pipelines.

From the variety of the supported by Virtuoso RDF formats, the Turtle syntax was chosen as the shortest and simplest way to describe objects and their properties. The main goal of the development was to implement a data converter for the given use case and to fill the Virtuoso storage with data for the follow-up tests and development, and this format looked the best choice – as it allowed one to implement the data conversion in the most straightforward and understood way. However, having some use cases for this type of transformation modules (and foreseeing even more of them), the authors also looked around for the way to create some more universal converter that would help us to avoid implementation of individual converter for every pipeline in the future work. Thus, the authors also looked on the other, more complicated RDF formats. One of them, JSON-LD (JSON for Linked Data), provides a possibility to describe data scheme along with JSON data or by referencing the description, stored as a separate Web resource. Since the inner dataflow pipeline format for transitional data is JSON, it looks very promising in terms of unification. Using JSON-LD one would be able to describe the general mapping of JSON keys, used in the transitional data, on the ontology objects and properties, and then upload JSON data directly, without additional transformation step. However, it would also impose additional restrictions on the modules producing data for conversion – at least, with the fixed set of conventional keys for the output data – and require extra work to adapt existing modules to fit them. For the given task it would be an irrelevant sophistication, but when it is decided that the Semantic Web approach proved itself for the DKB project, this more universal way for JSON to Turtle data conversion will be considered for the implementation.

3. Data processing pipeline

In the metadata integration process, each source repository can be linked to its own pipeline for data processing. Raw data from different storages often require several steps before they are ready for uploading to the final storage. The schematic view of one of such pipelines is presented in the figure 1. CERN Document Server (CDS) [10] contains public results of experiments in PDF files. The PDF content analyzer (Stage 3.0), processing the taken from CDS document, produces output data in JSON format. The last step in this chain (Stage 6.0) expects input data in the form of Turtle statements to upload them to the Virtuoso storage. Thus, the step between them (code name: Stage 5.4) was required to transform the output of Stage 3.0 into a Turtle format according to the general data scheme.

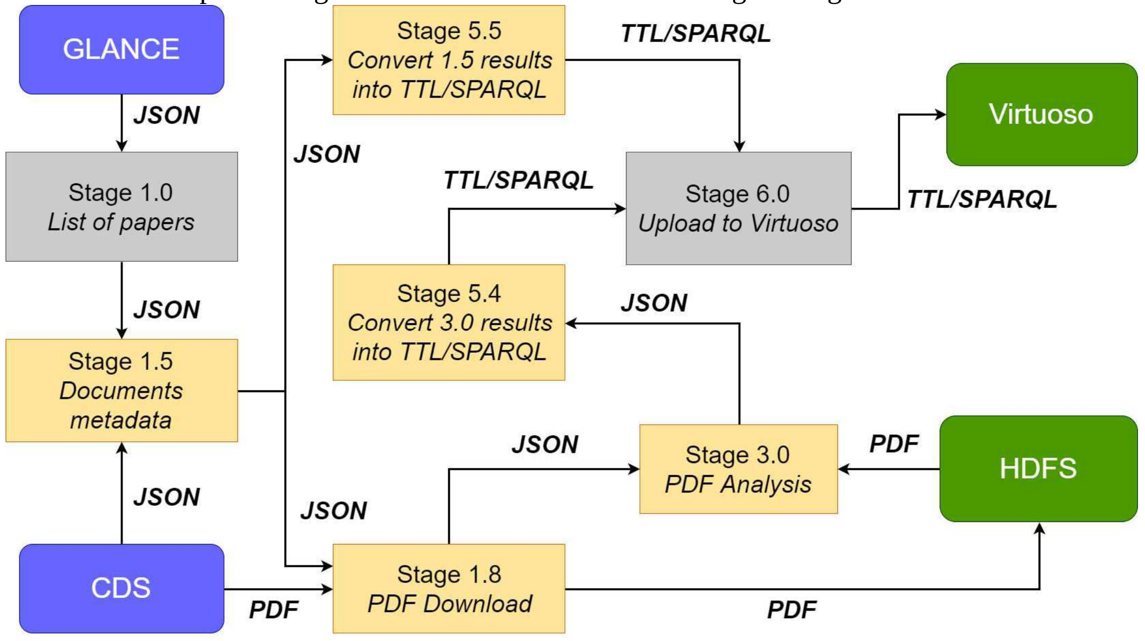

Figure 1. DKB Dataflow pipeline scheme.

The main task for the conversion module was to set a mapping of the given JSON structure to the ontology objects and their properties, and then to generate Turtle statements, semantically equivalent to the input data, according to this mapping. Also, as all the data transformation modules, developed for DKB integration subsystem, must obey a common set of rules, shortly addressed as a 'streaming mode' of the program. These rules were formulated so that any set of modules, supporting the streaming mode, could be joined into a pipeline by a supervising application that takes care of data transfer and guarantees exactly-once message delivery between modules.

In the streaming mode, the program should:

- consume input data from the STDIN;
- send output data to the STDOUT;
- send log messages to the STDERR;
- process one input message at a time;
- separate input and output messages with common markers;
- support common flow control specification (namely, inform the supervising program when a consumed input message is fully processed and the module is waiting for another one);
- operate in the infinite loop, waiting for input messages.

To simplify the development process for dataflow modules, a common library providing a standard implementation of input/output machinery was developed for Python language.

4. Evaluation

The developed module was tested on real data samples, produced by analyzing 200 papers from CDS at Stage 3.0.

After processing these samples at Stage 5.4, 4002 TTL statements were obtained in total. The average time of producing single TTL statement is 0.007ms, with standard deviation 0.02ms. As source data (PDF documents) widely vary in volume and complexity, the corresponding amount of metadata extracted at Stage 3.0 also varies from sample to sample. Because of these, the average processing time of single input message at Stage 5.4 (0.2ms) is less indicative; the time varies from 0.04ms to 5ms, depending heavily on the volume of the input data (which transforms to the number of produced TTL statements).

These results show that comparing to the other stages in this processing pipeline, Stage 5.4 has almost no influence on the overall processing time: at Stage 3.0 (the most time-consuming stage), average processing time is about 1 min, but in particular cases, it took up to 30 min for the given set of documents. So, the contribution of Stage 5.4 to the overall processing time is less than 0.0001%.

All tests were done using compute node with following characteristics:

- 48 GB RAM;
- 2 CPU(8 hyper-threading cores, 2.4 GHz).

5. Conclusion

The ETL modules for DKB can be developed individually, sharing only the input/output data and obeying a common set of rules. The module for data conversion from JSON to Turtle was developed following this paradigm and was successfully integrated into the automated dataflow pipeline from CDS to Virtuoso, allowing one to fill the RDF storage with data in an automatic way and go on with the work on the technology evaluation and conceptual architecture verification and improvement.

Despite the fact that the data conversion was implemented in the simplest possible way, appropriate only for the given use case, there is also a way for the unification of this kind of tasks. The approach with the JSON-LD format looks promising, and it will be implemented when needed for the further development of the DKB project [11].

6. Acknowledgments

The reported study was funded by Tomsk Polytechnic University Competitiveness Enhancement Program grant (TPU, CEP-IC-110\2017) and Russian Science Foundation under contract No.16-11-10280.

References

[1] Evans L, Bryant P 2008 *Journal of Instrumentation* **8(3)** S08001
[2] Cranmer K, Heinrich L, Jones R, South D M 2015 *Journal of Physics: Conference Series* **664** 032013
[3] Grigorieva M A, Golosova M V, Gubin M Y 2016 *Journal of Physics: Conference Series* **762** 012017
[4] Allemang D, Hendler J 2011 *Semantic web for the working ontologist: effective modeling in RDFS and OWL*, ed M E James (MA: Elsevier)
[5] ATLAS Collaboration 1992 *Letter of Intent for a General Purpose pp Experiment at the Large Hadron Collider at CERN* reports CERN/LHCC/02-04, CERN-LHCC-I-2
[6] Klein M, Ding Y, Fenset D, Omelayenko B 2003 *Towards the Semantic Web: Ontology-Driven Knowledge Management* **1** 47-70
[7] Klepmann M 2016 *Making Sense of Stream Processing: The Philosophy Behind Apache Kafka and Scalable Stream Data Platforms*, ed S Cutt (CA: O'Reilly) pp 39-100
[8] Milosevic Z, Chen W, Berry A, Rabhi F A 2016 *Big Data: Principles and Paradigms*, ed R Buyaa, R N Calheiros and A V Dastjerdi (Cambridge: Elsevier) pp 39-61
[9] Wang M, Liu J, Zhou W 2016 *8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2016)* **1** 363-368
[10] CERN Document Server URL https://cds.cern.ch/
[11] PanDAWMS/dkb: Data Knowledge Base for HENP experiments URL

https://github.com/PanDAWMS/dkb