2018 12th IEEE International Conference on Semantic Computing

Enhancing Big Data with Semantics: The AsterixDB Approach (Poster)

Wail Alkowaileet[*], Sattam Alsubaiee[†], Michael J. Carey[*], Chen Li[*],
Heri Ramampiaro[‡], Phanwadee Sinthong[*] and Xikui Wang[*]
[*]University of California Irvine
Email: w.alkowaileet, mjcarey, chenli, psinthon, xikuiw@ics.uci.edu
[†]Norwegian University of Science and Technology
Email: heri@ntnu.no
[†]Center for Complex Engineering Systems at KACST and MIT
Email: ssubaiee@kacst.edu.sa

***Abstract*—We explain how Apache AsterixDB, an open source Big Data platform, can help reduce the burden of extracting, storing, and exploiting semantics in Big Data. We describe its support for semi-structured data, Data Feeds, and user-defined functions (UDFs). Using these features from AsterixDB, users can easily create customized dataflows for semantic data analyses.**[1]

I. Introduction

Data is being generated every second, and extracting valuable insights from massive datasets has become more challenging than ever. Currently, data scientists have to maintain and glue together multiple heterogeneous platforms to be able to handle Big Data analytics tasks. To reduce the levels of expertise and the effort required for data scientists to work with Big Data, we need a system that provides scalability for data analysis on large-scale datasets, extendability with external machine learning libraries, and integrability with Big Data analytics tools. To meet this need, we use Apache AsterixDB [1] and enhance it with data analytics features. Apache AsterixDB provides support for scalable storage and analysis of large volumes of semi-structured data. It provides a flexible data model for enriching data with semantic information, data feeds for fast data ingestion, and a user-defined-function (UDF) framework for creating customized dataflows with external libraries. In this paper, we describe how to leverage these features for supporting end-to-end semantic data analytics.

II. Related Work

Most current approaches for end-to-end data analytics require gluing together multiple disjoint components. Platforms that provide similar functionalities include Hive [2], Spark [3], TensorFlow [4], and Weka [5]. One main limitation of these systems is, however, that they all need to be tied with other platforms to work with large datasets. Maintaining such systems demands computer system expertise from analysts who should instead focus on data modeling, selection of machine learning techniques, algorithms, and data exploration.

III. Our Building Blocks

A. Apache AsterixDB

Apache AsterixDB is an open source Big Data management system that provides distributed management of large-scale semi-structured data. Here we focus on several features that are important for data analytics.

1) User Model: AsterixDB provides a RESTful API for users to access data. Data scientists can use its query language, SQL++ [6], to access datasets without caring about low-level programming. The AsterixDB data model (ADM) is a superset of JSON and supports complex objects with nesting and collections. By defining a datatype to be "open," users can store semi-structured data with additional fields in AsterixDB without pre-specifying a complete schema.

2) Data Feeds: In many Big Data applications, data is generated continuously at a high speed. How to persist fast moving data efficiently is an important problem in building Big Data analytics applications. AsterixDB provides "data feeds" for ingesting/parsing/storing data from external sources with scalability and fault tolerance [7]. A user can ingest data from files, a prescribed socket, Twitter, RSS feeds, and other Internet sources by creating or using internal feed adapters and connecting them to AsterixDB datasets.

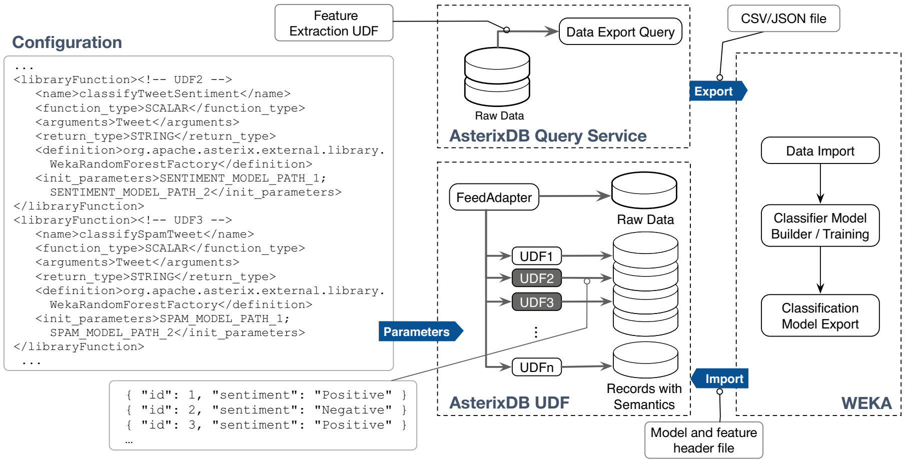
3) External User Defined Functions: There are cases where a user wants to perform complex operations on stored data not easily expressed in a declarative query language. AsterixDB provides an external user-defined function (UDF) framework to allow users to plug their own Java functions into AsterixDB. By implementing the UDF interface, a user can read and manipulate records in external UDFs. Once defined, UDFs can be used similar to native functions in queries.

B. Machine Learning Platforms

Machine learning algorithms are commonly used in learning models and extracting information from Big Data. Machine learning libraries come with a variety of usability, scalability, and extensibility characteristics. By utilizing the UDF framework discussed above, users can integrate their favorite libraries and use machine learning algorithms for complex data analytics tasks.

[1] The work was supported in part by NSF CNS award 1305430.

0-7695-6360-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICSC.2018.00063



```
...
<libraryFunction><!-- UDF2 -->
   <name>classifyTweetSentiment</name>
   <function_type>SCALAR</function_type>
   <arguments>Tweet</arguments>
   <return_type>STRING</return_type>
   <definition>org.apache.asterix.external.library.
     WekaRandomForestFactory</definition>
   <init_parameters>SENTIMENT_MODEL_PATH_1;
     SENTIMENT_MODEL_PATH_2</init_parameters>
</libraryFunction>
<libraryFunction><!-- UDF3 -->
   <name>classifySpamTweet</name>
   <function_type>SCALAR</function_type>
   <arguments>Tweet</arguments>
   <return_type>STRING</return_type>
   <definition>org.apache.asterix.external.library.
     WekaRandomForestFactory</definition>
   <init_parameters>SPAM_MODEL_PATH_1;
     SPAM_MODEL_PATH_2</init_parameters>
</libraryFunction>
 ...
```

```
{ "id": 1, "sentiment": "Positive" }
{ "id": 2, "sentiment": "Negative" }
{ "id": 3, "sentiment": "Positive" }
…
```

Fig. 1. Data analytics cycle with Weka and AsterixDB.

IV. Semantics in AsterixDB

A. The Life Cycle of Big Data

A common analytics workflow consists of the following three main tasks: (1) Data collecting, (2) Model training, and (3) Predicting. AsterixDB provides feeds for supporting rapid data ingestion (Task 1). In the following sections, we discuss how AsterixDB utilizes the building blocks in Section III to complete the other two tasks. Fig. 1 shows an end-to-end data analytics example using the Weka library.

B. Extracting Semantics from AsterixDB data

AsterixDB provides several different ways of serving data to different machine learning platforms. When a dataset is small, a user can query the data from AsterixDB and feed the results into a separate feature-extraction program, or they can embed feature-extraction code into a UDF, and query and export data features directly. For large datasets, AsterixDB can be coupled with a scalable machine learning platform such as Apache Spark. The AsterixDB-Spark connector [8] links the two parallel systems and has the ability to exploit data locality to minimize the data transfer cost whenever possible.

C. Incorporating Semantics into AsterixDB

With the UDF framework described in Section III, AsterixDB can utilize machine learning libraries to add/extract semantics to/from large-scale datasets. A user can choose to use a specialized library for a particular task, or use a general library with a trained model file. With the general library, users can reuse/rename the same UDF with different model files for different use cases. As shown on the left-hand side in Fig. 1, a user can reuse the same Randomforest UDF for sentiment analysis and spam detection by loading different model files. As part of our ongoing work, we are using similar mechanics to enable native Spark model evaluation in parallel on AsterixDB.

D. Data Scientist Sandbox

To further reduce the effort of working with multiple sources and engines for scientists, the notebook user interface has come into play. This interface helps scientists to be more productive by providing a unified interface for organizing and executing their code, and visualizing and exporting results without referring to the details of a low-level system. AsterixDB provides an implementation of its SQL++ interpreter [8] for Apache Zeppelin to help users formulate SQL++ queries in Zepplin for further analyses.

V. Conclusions

We have explained how Apache AsterixDB can be used in extracting semantic information from large-scale datasets. We introduced its data model, UDFs, and data feeds, and explained how they can work together with off-the-shelf machine learning libraries to provide a full end-to-end analytics experience for data analysts using their preferred tools. With this support for the loading-training-prediction life cycle in data analytics, we can provide users an integrated distributed Big Data platform with customizable dataflows.

References

[1] S. Alsubaiee *et al.*, “AsterixDB: A scalable, open source BDMS,” *PVLDB*, vol. 7, no. 14, pp. 1905–1916, 2014.

[2] A. Thusoo *et al.*, “Hive: a warehousing solution over a map-reduce framework,” *PVLDB*, vol. 2, no. 2, pp. 1626–1629, 2009.

[3] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster computing with working sets.” *HotCloud*, vol. 10, no. 10-10, p. 95, 2010.

[4] M. Abadi *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

[5] “Appendix b - the WEKA workbench,” in *Data Mining: Practical machine learning tools and techniques*, 4th ed., I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Eds. Morgan Kaufmann, 2017, pp. 553–571.

[6] K. W. Ong, Y. Papakonstantinou, and R. Vernoux, “The SQL++ query language: Configurable, unifying and semi-structured,” *arXiv preprint arXiv:1405.3631*, 2014.

[7] R. Grover and M. J. Carey, “Data ingestion in AsterixDB.” in *Proc. of EDBT 2015*, 2015, pp. 605–616.

[8] W. Y. Alkowaileet, S. Alsubaiee, M. J. Carey, T. Westmann, and Y. Bu, “Large-scale complex analytics on semi-structured datasets using AsterixDB and Spark,” *PVLDB*, vol. 9, no. 13, pp. 1585–1588, 2016.