Evaluation of Intuitive Platforms for Data analytics

September 2016

Author:

Denitsa Ivova Panova

Supervisor(s):
Antonio Marin Romero
Manuel Martin Marquez

CERN openlab Summer Student Report 2016

15 years CERN openlab

Abstract

Nowadays the world is over flooded with information. With the rise of Big Data, huge amounts of information, diverging in their variety, velocity and volume, a new data analytics branch has developed. It has emerged from the need to offer to the researchers an easier way, an intuitive software with which to analyze the accumulated pile of information.

The purpose of this project is to investigate some of these data analytics platforms and evaluate them for the unique requirements of CERN environment and use cases. The question which needs to be answered is - are those platforms able to facilitate the users to conduct analysis without the necessity to acquire new skill set. Initial point of the report is to give context of the project and to specify why the use of Big Data technologies is required at CERN. Additionally, it gives an overview of the Big Data tools which are used in the evaluated platforms. The core of the report consists of the mechanism behind each of three data analytics platforms and their evaluation in terms of their merits and downsides.

Table of Contents

1 Introduction

What is Big Data? That question has been asked frequently recently. If we were in third century BC, then the Alexandrian library would have been accounted as the biggest source of information, the house which encloses all human knowledge under one roof. Today most people on the planet have access to 320 times more information than what it is believed the library to hold. So what we can say Big Data is? One Alexandrian library? Two Alexandrian libraries?

An exact definition is hard to be given since our notion of "big" stretches every day. Nowadays, one-quarter of the world's information is digitalized and just less than two percent of all stored data has not been digitalized yet. The quantity of the available data is doubling every year which poses the following question - what should we do with it? The data which is acquired has the key feature to differ immensely in its velocity, volume and variety. Therefore, analyzing it or extracting features is a task which has been arduous before the rise of the Big Data tools.

Today the market is swirling with Big Data tools and they all try to make the data exploration easy and fast, for example such tools are Hadoop or SQL. Moreover, there are a lot of platforms which are built on top of those tool in order to facilitate easier usage since working with them requires certain skill set. Microsoft has recently deployed a web-based application which is an excellent example of good data analytics platform - Microsoft Azure. It is very intuitive and simple to use. The purpose of this project is to evaluate three different platforms in the CERN context and to assess how useful they are for the CERN users.

Before discussing the merits of those platforms, we answer the following questions - what is big in the CERN data? Why do we need to consider utilizing such tools as a whole? Every second around 600 million events recorded from every experiment. From them 100 000 are sent for digital recognition and from those, only 100-200 are classified as interesting. Even though the dimension of the problem has been reduced immensely, the data that is stored for analysis accumulates faster than we can process it. Every day the data is stored in the Data Center and every second a 1.5 CD is burnt - amount that definitely can be classified as big data [5]. Furthermore, according to Bob Jones, Project Leader at CERN, "research moves quickly." and "physicists develop and change focus in their experimental work" rapidly [1]. Therefore, the utilization of data analytics platforms which can handle large amounts of data is essential so that the analysis keeps up with the theories that need to be tested.

As mentioned before, the goal of this report is to investigate three different data analytics platforms and evaluate their relevance for CERN: H2O, PredictionIO and Oracle Stream Analytics. The report structure is as it follows - initially we begin by defining some tools which stand behind the data analytics platforms which we discuss, then we dedicate a subchapter for each of the platforms, providing a low-level overview and short cost-benefit analysis.

2 Background Information

The platforms are built on top of Apache Hadoop and/or Spark. Therefore, before we start introducing them, we will give quick overview of both pillar technologies.

Apache Hadoop

Apache Hadoop is an outcome of the increasing demand to handle the endlessly accumulating data. The idea was born from the Google File System paper (2003). The grounds for this publication was that the web generated more and more data every day and it was getting harder and harder to index all the pages.

In its essence Hadoop is a very good storage distributed across several clusters and having the ability to run "distributed" tasks on every of those clusters. Apache Hadoop is open-source software for reliable, scalable, distributed computing. Those key characteristics are achievable due to its two linchpins: data processing framework (MapReduce) and a distributed storage system (Hadoop Distributed File System).

Hadoop is not a database and we cannot use SQL language to access data. It more or less resembles data warehouse, thus it can store both structured and unstructured data. The way to access the information from the storage system is by the means of MapReduce which runs series of jobs. MapReduce is highly flexible since it can pull information from both structured and unstructured source, yet it is also complex to use. Today there are a lot of tools which facilitate the communication to MapReduce, for instance Hive which translates SQL commands into MapReduce Jobs.

Another key feature for Hadoop is that in the cluster, the data within HDFS and the MapReduce system are located on every machine which makes the platform reliable to use in case one machine in the cluster goes down. It also means that we can add as many new machines as necessary without the need to adjust the previous components in the Hadoop cluster providing scalability feature of the tool. Having in a very machine HDFS and MapReduce system allows for distributed processing of larger and larger data sets since every new machine contributes additional hard drive space and processor power. Therefore, having the resources, there is no limit of how big the data can be.

Since the cluster can continue to work even if one machine is not functional anymore, data loss should be considered. Such failures are taken into account. This is done by separating the large data file into small blocks of data. Than each block is replicated three times and stored and processed in different machines, a measure which ensures reliability.

Having all of the positive feature of Hadoop in mind, what are the challenges of using it? Although MapReduce has innovated the big data processing, it is not good match for all problems. In order for the MapReduce to be applicable measure, the task at hand needs to be parallelized. Thus, problems which are iterative are not appropriate.

Spark

Apart from Hadoop other Big Data frameworks is Spark. It provides some of the most popular tools used to carry out common Big Data-related tasks. The benefit which Spark provides is speed. It is reported to be 100 times faster than Hadoop [2]. Yet, its downside is that it does not have its own distributed system, therefore, it is usually on top of another one like HDFS.

Another important distinction between the two frameworks is how they handle the operations. Spark runs most of its operations "in memory" – copying the data from the distributed physical storage into the far faster logical RAM memory. This reduces the amount of time. On the other

hand, MapReduce, or Hadoop, writes all of the data back to the physical storage medium after each operation. This was originally done to ensure full recovery in case something goes wrong. Yet, it slows down the process.

What about in terms of machine learning and data analytics? Do the platforms differ in that aspect, too? Indeed. Spark includes its own machine learning libraries, called MLlib, whereas Hadoop systems must be interfaced with a third-party machine learning library, for example Apache Mahout.

3 H2O

The first Data Analytics platform which we evaluate is H2O. It is actually a library, therefore, we compare it to MLlib, the native library for Spark. Both of them are available in the same environments - R, Python and Spark. Moreover, the algorithms which both provide overlap to great extend. What about differences? One of the main ones is that H2O is created by a single entity Oxidata, whereas MLlib is community driven. Therefore, H2O is more consistent in terms of the arguments and the features of the different algorithms. This makes H2O much more intuitive to work with. Another distinction is related to the data format. H2O can work only with H2OFrame which creates inconvenience for the user since the data should be transformed constantly. Moreover, each data transformation risks mistakes in how the data is translated to the new format. On the other hand, MLlib is much more flexible regarding the data format. It can work with Resilient Distributed Datasets, DataFrames and DataSets.

A key feature of H2O is its speed. The following graph is from an independent report which compares different machine learning libraries [8]. The juxtaposition is conducted using linear models. The x-axis indicates the number of rows in the examined data set. It is important to be noted that the scale is 1:1000000. Additionally, the y-axis shows the time for execution in terms of seconds.

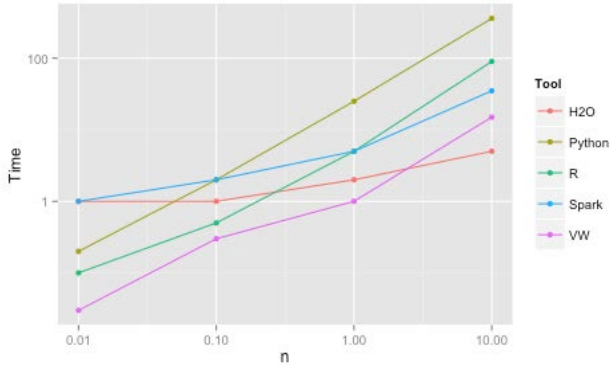

Figure 1: Comparing execution time using linear models [8]

In the graph MLlib is marked as Python. The speed difference is more easily recognizable when the data set is large, if the data exceeds 10000 rows.

H2O can be deployed on Hadoop cluster in order to be the third-party library which we discussed in the previous chapter. We have managed successfully to install it on a cluster of 4 win-large nodes with 8 GB ram memory. An important prerequisite is that the firewall is disabled. For Centos7 the relevant command is:

$ disable SELinux

Another key step which is not mentioned in the official documentation and is crucial for deployment is:

$ export HADOOP_USER_NAME=hdfs

Otherwise, installing H2O on a cluster is straightforward and the official documentation is easy to follow.

Another relevant question for evaluating H2O is if the library is intuitive and user-friendly. The following snippet of code shows how to implement Gradient Boosting Machine algorithm in R.

```
> y<-"Class"
> x<- setdiff(names(train),y)
> fit<-h2o.gbm(x=x, y=y, training_frame =train)
> pred <- h2o.predict(fit,test)
```

Executing the algorithm on artificially created data set proves that it is easy to use the package. Yet, another question which needs to be posed is it developer-friendly, or put in other words, is it simple to alter the algorithm. Although H2O is open-source project and the source code is available in GitHub repository, it is not intuitive to change it.

Another product which H2O team will offer soon is Steam. It allows developers to make predictions using a simple REST/RPC service and data scientists to deploy and publish predictive models. This product sounds promising and a future step in accessing the H2O.ai would be to evaluate this extension of the platform.

4 PredictioIO

The second platform which I evaluated is PredictionIO. In its essence it resembles the idea of H2O Stream which we mentioned before. PredictionIO is an open source machine learning server which requires to be installed on a local machine. It is built over the DASE architecture. DASE stands for Data, Algorithms, Server and Evaluation.

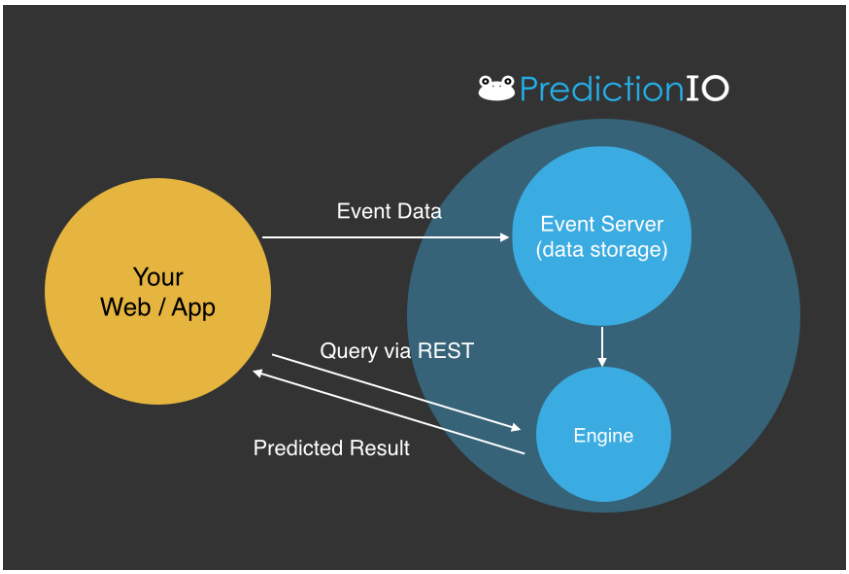

Figure 2: PredictionIO Architecture [7]

The platform works as it follows - data is collected from your Web application and it is stored in the Event Server. Than the information is requested from the Engine which stores the chosen algorithm or ensemble of algorithms. In the Engine the algorithm is trained and real-time predictions are generated. Then the accuracy of the algorithm is evaluated by comparing the predictions to the real-time data. At the same time the algorithm is retrained based on the evaluation score.

Advantage of the platform is the stack of machine learning templates which the team provides. They are advertised to be easy to alter and personalize to the user's' data set. Yet, the templates are solely written in Scala which makes the tool language inflexible. To use a template is straightforward. The following shell commands summarize the main steps:

First we download the template

$ pio template get <template_repo> <your_application_directory>

We initialize the new application

$ pio new app MyApp1

Start building the application

$ pio build

Start training the application with the data which we have initially uploaded

$ pio train

In order to explore the platform we have deployed several versions. The first one which we installed was PredictionIO 9.6. It is supposed to be the most stable version. The automatic deployment (Quick Start) which the team offers is successful. Yet, when we start training the algorithm, there are no predictions available. It produces an empty prediction array. After investigating the issue, we have found that it is due to a problem with the most current update of the release. Therefore, we decided to deploy the PredictionIO 9.7 version manually. Even though all dependencies have been installed successfully (Elasticsearch, Hadoop and Hbase), running it has not been possible because a file was missing.

Although the PredictionIO has been advertised as an easy to use platform, it is impossible for us to access its merits due to the deployment issues. The PredictionIO team needs to amend the glitches in order for users to be able to benefit from their product.

5 Oracle Stream Analytics

The last data analytics platform which we have accessed is Oracle stream analytics. It is web-based application which has intuitive user interface. To conduct data analysis, a user needs to push several buttons.

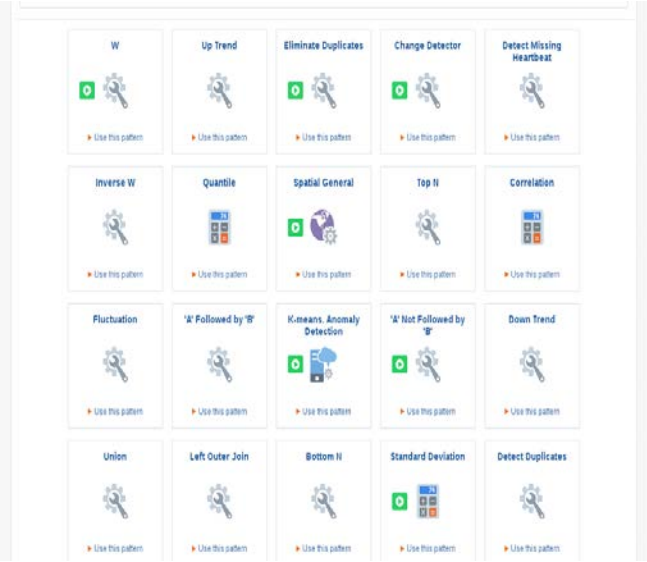

Figure 3: Oracle Stream Analytics Library Screenshot

It offers machine learning algorithms, for example K-means clustering. Additionally, we can work with batch and stream data and on top of that it provides real-time statistics of the stream data, as seen in the picture below.

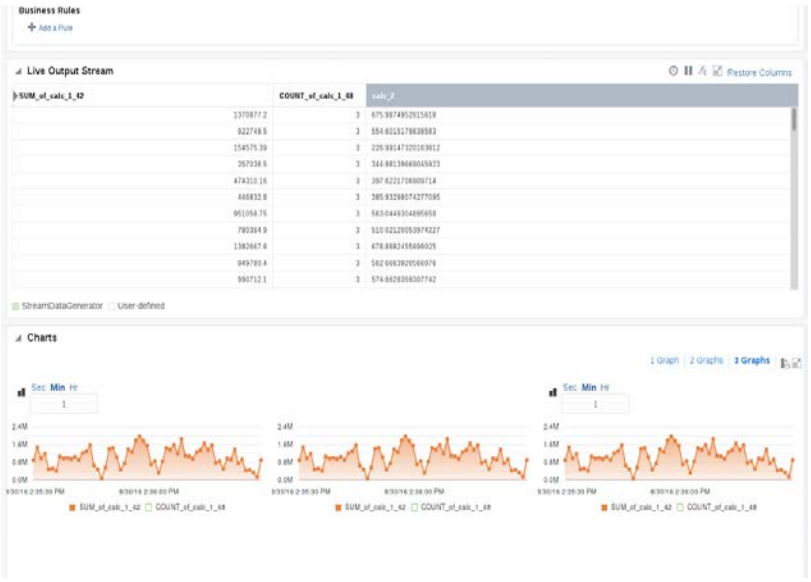

Figure 4: Oracle Stream Analytics Real-time Statistics Screenshot

While exploring the platform, we found an issue regarding the naming of local variables. It is not possible to name them with SQL reserved words but the input is not sanitize in the web application at the moment which can produce runtime errors. Another issues were found trying to work with Avro data. Based on the results of this evaluation, a report will be shared with the Oracle Stream Analytics team within the openlab collaboration framework. This report will include detailed information and feedback about the evaluation as well as desired additional functionalities that could be implemented in future releases.

A possible extension of the Oracle Stream Analytics platform evaluation would be to connect the software to sensor data from the Power Converter systems for the CERN accelerator. The goal would be to generate summary statistics. As a proof of concept we evaluate root mean square using artificial data. The following steps indicate the procedure:

- We simulate stream data with two floating variables - value1 and value 2

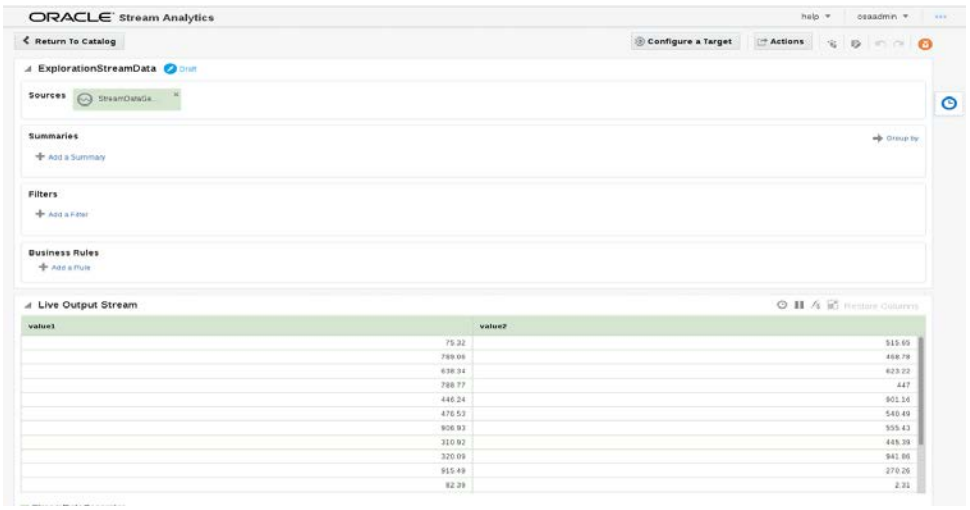

Figure 5: Oracle Stream Analytics Data Screenshot

- We focus mainly on the first variable. We calculate its square.

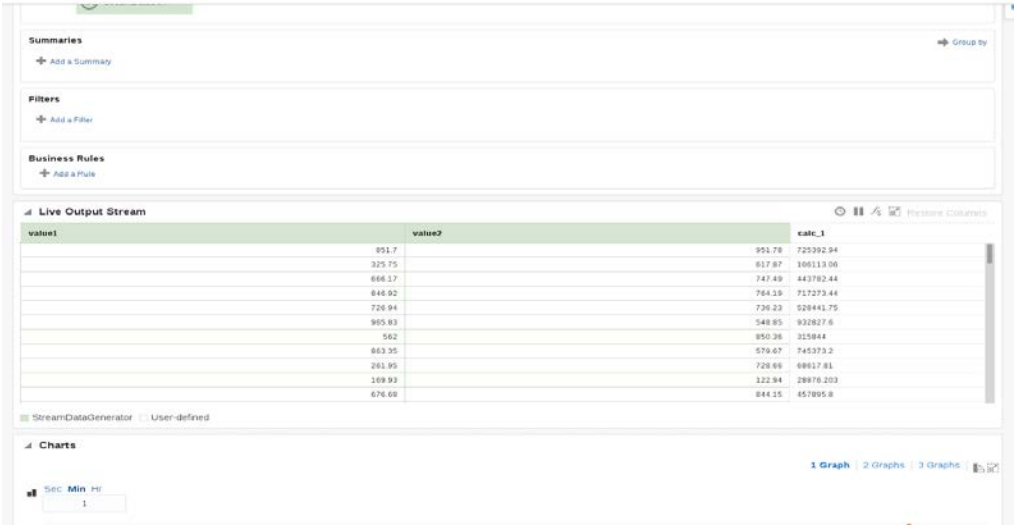

Figure 6: Oracle Stream Analytics Calculating the Square Screenshot

- As a final step we estimate the root mean square statistics by summing and counting the data entries every 10 seconds

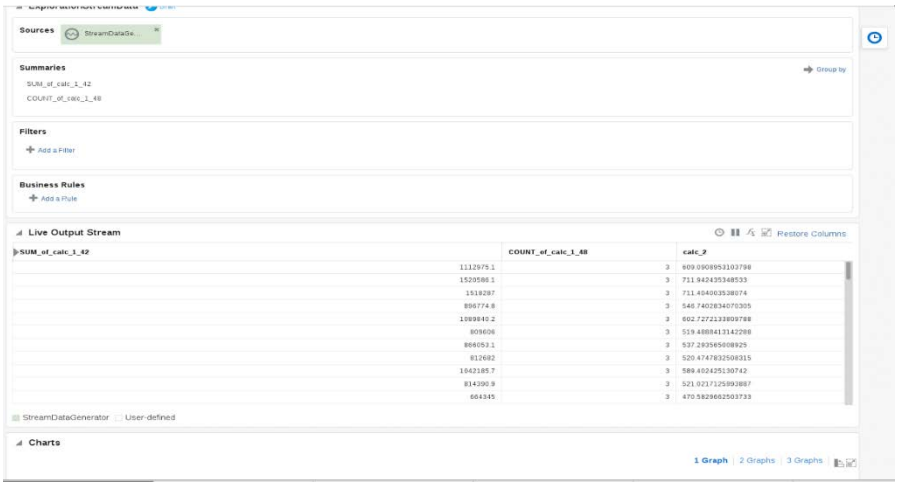

Figure 7: Oracle Stream Analytics Calculating Root Mean Square Screenshot

6 Conclusion

CERN, as one of the major worldwide research institutes, produces huge amounts of data. To be precise the Data Center processes about one petabyte of data every day. In order to effectively and efficiently to manage the data and conduct analytics, we need the help of Big Data software. In this report we have evaluated three such data analytics platforms. The first one is a library which can be used in Spark, Python or R environment - H2O. It strikes with its speed of executing machine learning algorithms even if the data set exceeds 10000 raws. Although it provides user-friendly commands, it is not developer-friendly since it is not easy to alter the provided algorithms. The second platform is PredictionIO. It is a server which offers a lot of easy-to-implement machine learning templates. Yet, the platform is impossible to be deployed at this point. The last tool evaluated is Oracle Stream Analytics. This web-based application has intuitive interface and can provide real-time statistics of steam data which can be very useful for certain use cases where fast processing of the data is required.

7 References:

[1] Cukier, Kenneth Neil, and Viktor ViMayer-Schoenberger. "The Rise of Big Data." Foreign Affairs. N.p., 3 Apr. 2013. Web.

[2] Marr, Bernard. "The Big 'Big Data' Question: Hadoop or Spark?" N.p., 19 July 2015. Web.

[3] "Hadoop and Big Data." Mapr. N.p., n.d. Web

[4] "This Is What Big Data Really Looks Like: CERN, the Universe and Everything." CLoud Watch Hub. N.p., 6 Aug. 2015. Web.

[5] "Prcessing: What to Record?" CERN. N.p., n.d. Web.

[6] Hadoop. N.p., n.d. Web.

[6] H20.ai. N.p., n.d. Web.

[7] PredictionIO. N.p., n.d. Web.

[8] "Simple/limited/incomplete benchmark for scalability, speed and accuracy of machine learning libraries for classification" , n.d, Web.