December 2021

Databloom Blossom - Federated Analytics v1.0

Part one: Medical and Healthcare

Dr. Jorge Arnulfo Quiané Ruiz
Alexander Alten-Lorenz

White Paper

In this white paper we will show how to overcome today's challenges in Big Data and Global Analytics with Databloom.

Content

01 The Five Challenges of Data Analytics

02 The Federated Data Lakehouse Analytics Platform

03 Analytics in Medical and Healthcare - a real Story

04 Technical Overview

The Five Challenges of Data Analytics

More and more applications produce large amounts of diverse data. We currently have a plethora of data processing platforms, ranging from relational databases to machine learning platforms. Each of these platforms excels in different aspects of the design space. For instance, while a relation database will allow users to perform different analyses of the data, a machine learning platform will allow users to build e.g. predictive models.

There is an increasing interest from organizations to work together to solve complex problems. Typically, this requires them to analyze the data that is stored on their premises. However, data is often privacy sensitive and cannot be moved to a central place. There is thus a need to perform data analytics pipelines in a federated fashion: raw data stays where it belongs to and only aggregated data is sent out of data owners' premises.

Despite all advancements in data processing platforms, users typically will end up running their data analytics in a suboptimal way. This is because users are typically faced with five challenges, namely:

1) integrating multiple data sources (data lake);

2) selecting the right data processing platform among the myriad of big data platforms (zoo of systems);

3) unifying heterogeneous analytics tasks into a single data pipeline (unified data pipelines);

4) choosing the right cloud provider to run their data analytics (heterogeneous cloud providers), and;

5) more and more users are willing to collaborate to solve more complex problems (federated data analytics).

Data Lakes

A data lake is a storage repository that holds a vast amount of raw data in its native format until it is needed for analytics applications.

Organizations typically store their data in data lakes as their different departments produce data in different formats. This leads the organization to build ETL tools to bring data into a single “repository” with a common “schema”; to enable the analytics teams to analyze the data. This process is often expensive and hence must be avoided. Therefore, data must be accessed / analyzed from where it is produced.

Zoo of Systems

When analyzing data, one faces a plethora of options of data processing platforms that can do the job. Selecting the best platform depends on both the data and the data analytics tasks. This is a daunting task for users that most of the cases lead to suboptimal choices. Users must be relieved from this task.

Unified data pipelines

Today’s data analytics pipelines are moving beyond the limits of a single data processing platform as they are composed of several heterogeneous tasks, such as data preparation and machine learning tasks. The current practice is to divide data analytics pipelines into multiple dependent tasks, where each runs on a different data processing platform. This results in expensive data processing pipelines composed of several “silos” of data processing. Instead, data processing platforms must work in tandem when executing data analytics pipelines.

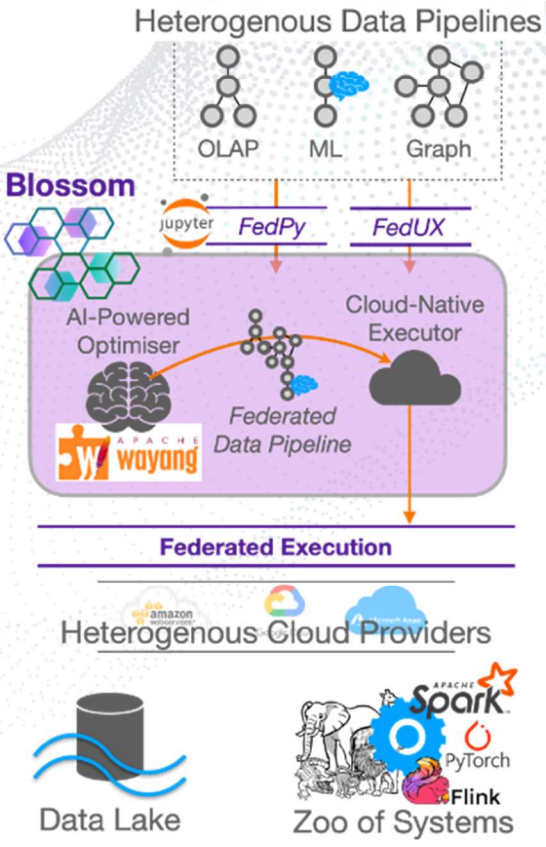
Heterogenous cloud providers

More and more organizations are moving their data analytics pipelines to the cloud. Yet, choosing the right cloud provider is far from trivial as it depends on the data analytics pipelines themselves: either you are trying to maximize performance or minimize costs. As data analytics pipelines are quite heterogenous, sticking to a single cloud provider might not be a wise decision. Organizations must be able to easily deploy each of their data analytics pipelines to the best cloud provider according to the organizations’ needs.

Federated Data Lake-House Analytics

Databloom Blossom is the unique Federated Data Lake-house AI and Analytics platform to tackle all these challenges at once.
Users now can focus solely on the logic of their applications and do not need to deal with infrastructures, job execution, and deployment details.

Simple, yet powerful, user interfaces. Blossom provides two simple interfaces for users to develop their pipelines: Python (FedPy) and a graphical dashboard (FedUX). While FedPY targets data scientists, FedUX targets developers in general. Users provide a data analytics pipeline specified in any of these interfaces and Blossom runs them on any cloud provider and data processing platform.



The code example shows the simple, but famous WordCount application in Blossom. The first three lines allow the user to register the platform to use in Blossom (Java program and Spark in our example). The remaining lines of code are the actual WordCount program.

```
// create a context and set the platforms
val wayangContext = new WayangContext(new Configuration)
  .withPlugin(Java.basicPlugin)
  .withPlugin(Spark.basicPlugin)

// create the PlanBuilder
val planBuilder = new PlanBuilder(wayangContext)

//Wordcount plan and the result in a list
val wordcounts = planBuilder
  .readTextFile(inputFile)
  .flatMap(_.split("\\W+"))
  .map(word => (word.toLowerCase, 1))
  .reduceByKey(_._1, (c1, c2) => (c1._1, c1._2 + c2._2))
```

The beauty of Blossom is that the user does not have to decide on which data processing platform to run the program (Java or Spark). Blossom decides the actual execution based on the input dataset's and processing platforms' characteristics (such as the size of the input dataset and the Spark cluster size).

The Blossom Data Platform is the federated AI/ML training and data analytics powerhouse. We enable Federated Data Lakehouse Analytics and Model Training across multiple data lakes and warehouses, independently of their origin.

Databloom's Blossom comes with state of the art technology:

AI-powered cross-platform optimizer

At its core, we can find Apache Wayang [1], the first cross-platform data processing system. Blossom leverages and equips Apache Wayang with AI to unify and optimize heterogeneous (federated) data pipelines as well as to select the right cloud provider and data processing platform to run the resulting federated data pipelines. As a result, users can seamlessly run general data analytics and AI together on any data processing platform.

Cloud-native and cross-platform executor

Blossom also comes with a cloud-native executor that allows users to deploy their federated data analytics on any cloud provider and data processing platform. They can choose their preferred cloud provider/data processing platform or let Blossom select the best cloud provider (data processing platform) based on their time and monetary budget. In both cases, Blossom deploys and execute users' federated pipelines on their behalf.

Federated execution

The AI-powered cross-platform optimizer and cloud-native executor of Blossom also take care of any data transfer that must occur among cloud providers and data processing platforms. While the optimizer decides which data must be moved, the executor ensures the efficient movement of the data among different providers and data processing platforms. Blossom provides users with the means to easily develop their federated data analytics in a simple and fast execution.

[1] https://apache.wayang.org

Story: Federated Analytics - Kidney Disease Research

Background

Hospitals and medical organizations are increasingly using machine/deep learning (ML/DL) to learn diagnosis and prediction models from medical data. In this success story, a US hospital research center trained a prediction model for chronic kidney disease. ML/DL have helped hospitals and medical organizations to better schedule surgical operations as well as to better predict patient re-hospitalizations after an organ transplant.

Technical Considerations

Most ML/DL algorithms require huge amounts of training data to achieve high accuracy. Therefore, building ML or DL models within the premises of a single research facility (i.e., on small datasets) typically leads to low accuracy. This is because small medical datasets do not contain a large population of patients, making them not diverse enough to build robust ML/DL models.

As a result, more and more hospitals and medical organizations desire to collaborate to build more robust models by putting their datasets together. Nevertheless, health data is incredibly sensitive and cannot be shared outside the premises of the data owner. This has been one of the main blockers in the fight against COVID-19: hospitals and medical organizations are willing to collaborate but all the legal framework around data privacy prevents them to share their patient's data.

Solution

Databloom' Federated Data Lakehouse Analytics (Blossom) platform is ideal for such settings. It uses a federated learning approach to bring the computation to the data instead of the data to the computation. In our healthcare story, Blossom allowed the research center, and therefore hospitals and medical organizations, to locally build the models and exchange their models among them to combine them into a single model. With Blossom, the team was able to plan transplants and care accordingly, which resulted into much higher transplanted organ adoptions.
Notice that Blossom does not move raw data but aggregated data only. The beauty is that users do not have to worry about the low-level details for moving models among different medical entities and combine them into a single model: Blossom does all this transparently and at scale.

User Interaction

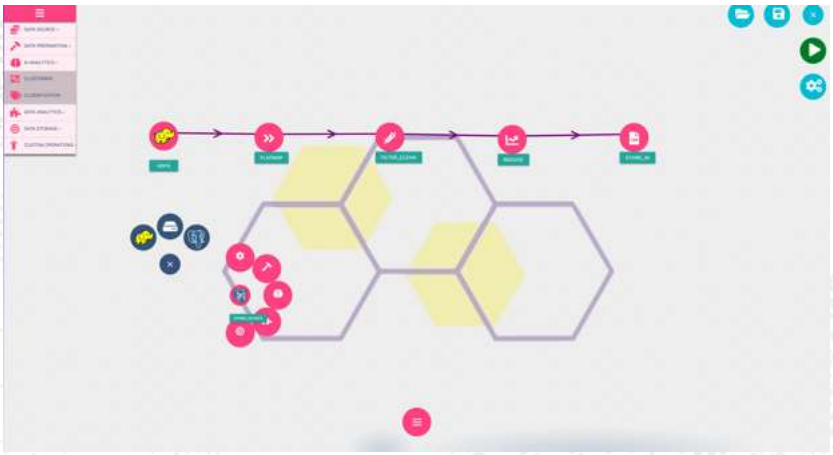
The Blossom Data Platform comes with an intuitive user interface (UI). This interface allows non-technical personnel to quickly create complex data pipelines without considering the underlying data architecture.
Compared with other platforms like Apache Spark or their commercial variants, organizing and managing federated analytics with Blossom is just a few clicks away.

Example Workflow

We cannot display the kidney disease research workflow, but we have built a data analytics flow based on the word count example we used in the code a few pages before.
The raw data comes from a HDFS (Hadoop) based data lake, and we want the output stored as a CSV file, ready to import into databases like postgres or other columnar data structures. The workflow would look like:

```
source -> flatmap() -> filterclean() -> reduce() -> output as CSV
```

This setup would take at least up to 10 minutes for an experienced data ops using available tools on the market. With Blossom it takes less than one minute, since the technical complexity is complete abstracted in our platform.



The user has just to “drag and drop” the available data lakes and combine them with the desired data mechanics. The platform takes care about the in-memory handling, data splitting and which framework is used to complete the task. Blossom takes care about the “translation” between used languages and frameworks, be it Java Streams, Python, Scala, Java or tensorflow.
As for example, Blossom translates Hadoop workflows, be it MapReduce() or even Hive, into modern and faster frameworks like Apache Spark or future computation frameworks, like quantum computing related tasks. This enables data scientists to concentrate on the given problem instead to deal with older or future platforms.

Technical Overview

Setup and Integration

Integrating Blossom into existing data lakes and data lake houses takes less than 4 hours, thanks to our cloud-native kubernetes operator. Blossom takes care about the data splitting and comes with his own parameter server, optimized for federated AI workloads.

Comparison Blossom ./. Big Data Frameworks

	Hadoop	Databricks	Presto	Blossom
Python support	Generic	Generic	Generic	Natively
Cross-Platform Optimization	--	--	Basic	Advanced
Multi Cloud Execution	--	--	--	Yes
Federated Execution	--	--	Basic	Advanced
Integrated Data Debugger	--	--	--	Yes

Supported Systems

- Apache Flink v1.7.1
- Apache Giraph v1.2.0-hadoop2
- GraphChi v0.2.2 (only available with scala 11.x)
- Java Streams (version depends on the java version)
- JDBC-Template
- Postgres v9.4.1208 (Implementation JDBC-Template)
- Apache Spark v3.1.2 (scala 12.x) and v2.4.8 (scala 11.x)
- SQLite3 v3.8.11.2 (implementation JDBC-Template)
- Hadoop 2 Map/Reduce

Online Ressources

https://databloom.ai
https://www.capterra.com/p/241081/Blossom-Federated-AI-Platform/
https://github.com/databloom-ai
https://wayang.apache.org/documentation/
https://github.com/apache/incubator-wayang

Scalytics

The Federated Data Company

www.scalytics.io

Scalytics, Inc.
3401 N. Miami Ave. STE 230
33127 Miami, FL.
United States