Big Data Technologies

Eric Solano, Ph.D.

Big data technologies I have used and gained expertise in include Apache Spark, Amazon Redshift, Apache Mahout, Apache Cassandra and the Hadoop Ecosystem, with focus in HDFS.

Continental manufacturing data has very high dimensionality. The tire manufacturing process involves several steps: Design, Mixing, Extrusion, Molding, Vulcanization and Inspection. All these processes collect large amounts of data from equipment and sensors at the manufacturing plant and store the data in a system called DOPAC.

One of the projects under my leadership used data sources from one of Continental plants located in Korbach, Germany. The dataset has multiple data sources from 2020 through 2023. For example, the Vulcanization dataset contains data from 14 different machines, with a total of 1.1 billion records.

Big data technology was required to analyze these datasets. I used the Apache Spark platform, the Hadoop Distributed File System (HDFS) and the Apache Mahout framework. An Amazon EMR cluster with 5 nodes was used. Each EMR cluster node is of type 'm5a.xlarge' (4 CPUs, 16 GB RAM). This architecture allowed for the analysis of distributed big data in a fraction of the time it would require for the same datasets to be processed on a single node. Data profiles for all datasets were performed using Spark Scala and pySpark scripts. All datasets are high dimensionality datasets. Processing times in Spark are quite small for the magnitude of the datasets. I used Regression Analysis to explain Curing time using a few independent variables from the dataset. I used Gradient Boosting Trees (GBT) algorithms from the Spark ML library.

At Continental, I oversee the continuous development and maintenance of the Telemetry Backbone (TBB). The TBB's main purpose is to support Data Scientists with a solid and easily accessible layer of all available and enriched telemetry data. With the emergence of the Internet of Things, large volumes of data are generated today. The TBB with its underlying architecture is meant to provide Continental telemetry data of any kind. It holds data from different loggers (e.g., Questar Auto Technologies, FlexBox, Avisaro, etc.) and aims at providing data enrichments such as CRM Master data or data from other providers such as altitude or weather conditions for GPS data. Data is usually provided in its initial aggregation (usually transactional level) to allow for better analysis.

The TBB uses Apache Cassandra as the persistence layer (storage). Data is currently being ingested via stream (mainly from the ContiConnect Kafka Broker) or via batch process from different batch engines. The TBB uses a message broker (Apache Kafka) and provides dedicated topics per data source as ingestion zone for incoming telemetry data.

I implemented an access layer to the TBB using the DataStax Python Driver for Apache Cassandra. This driver works exclusively with the Cassandra Query Language v3 (CQL3) and

Cassandra’s native protocol. The TBB python bindings use the DataStax API to connect to the TBB Cassandra cluster, and to create an Authenticator class instance with the necessary credentials.

To take advantage of both Apache Cassandra and Apache Spark platforms, I installed the Spark platform collocated with the Cassandra cluster. I have experienced several advantages to using Apache Spark and Cassandra together on the same cluster: data locality, scalability, high availability, key-value data modeling, integrated analytics and high performance.

In the last couple of years, I have been engaged with Artificial Intelligence and Large Language Models (LLM) activities at Continental. At Continental, for generic access and usage of LLM, there are a few tools already available: an AI Assistant based on the Microsoft Azure OpenAI Service; Microsoft Copilot for Web; GitHub Copilot which uses prompts and pulls snippets of code to create good suggestions for coding needs. I am also sharing and exploring additional customized generative AI solutions: usage of closed-source models, such as GPT-4 or Claude, via providers like Microsoft or AWS; usage of open-source models, such as Llama3 or Mistral. With strictly confidential data, our own infrastructure is required, and open source is the only option. With public, internal or confidential data, using a provider is accepted and "pay as you go" is more cost-efficient.