HAKOM Time Series GmbH

Simeon Harrison (NCC Austria),
February 11th, 2022

simeon.harrison@tuwien.ac.at

Industrial HPC Course

SUCCESS STORY IN TIME SERIES ANALYTICS

TEMPLATE 1 (A)
Overview

HAKOM

HAKOM is the technology leader for time series management in the energy industry. It deploys a uniquely simple time series technology with outstanding performance.

THE PROBLEM

- Increased demand for high speed and high volume data analytics.
- Running Time Series Manager in a cluster environment

SUCCESS STORY DETAILS
HPC provider: LBD
Domain Expert: Giovanna Rhoda
Country: Austria

THE HPC PROBLEM DOMAIN

- Writing highly parallelized ETL scripts
- Sudden ingress of big data into a classical relational database.
- Loading the dataset with Spark's JDBC connector

THE SOLUTION

- Exporting time series data to parquet files
- Loading the data into HDFS
- Configureing and deploying Apache Spark worker packages
- Running analytics on TU Wien's Little Big Data cluster

THE BENEFITS

- Decreased loading time of dataset
- Decreased computation time for analytics
- Experience in running a problem on a cluster

Design inspiration NCC Austria

EURO

SUCCESS STORY IN TIME SERIES ANALYTICS

TEMPLATE 1 (B)
Additional Details

THE PROBLEM

As many other sectors enter the digitalization era, HAKOM is bracing for data management and analytics problems that demand a much higher volume and processing speed than ever before. Such requirements from the market have motivated our team to investigate new base technologies to add more advanced big data analytics capabilities to the repertoire available to HAKOM's customers.
The goal of this project was to classify signal data into gravitational waves or noise. For that goal, the data from three different detectors needed to be processed and set as an input for Machine Learning models.

THE HPC PROBLEM DOMAIN

Ingress of big data into a classical relational database:
The ingress stayed within the usual performance characteristics of an operational system, which is too slow for loading very large analytical datasets. The loading times for the training set series data ranged from multiple hours to up to a day.
Loading the dataset using Spark's JDBC connector also proved to be inadequate for this use case. We explored the loading time of the distributed data frames from very small amounts of time series to the complete training set and experienced extremely quick degradation of load performance. In both cases the reliance on the single relational database appeared to be the bottleneck.
In both cases the reliance on the single relational database appeared to be the bottleneck.

THE SOLUTION

We explored the capabilities of TU Wien's Little Big Data (LBD) Cluster on the basis of a Kaggle G2Net data set within a single, focused analytical process.
With the guidance of TU Wien's DataLab team we decided to export our time series data to parquet files holding three different sensor readings and loading them into HDFS. The initial extraction, transformation and load (ETL) process took some time but was significantly faster than the status quo.
With all the data residing in the distributed network accessible file system reading subsets of data – roughly 10 GB – into a Spark distributed data frame became both doable and reasonably fast.

THE BENEFITS

In the end we were able to find a combination which performed favorably and construct a deployable artifact, that was able to yield descriptive statistics for the whole dataset in roughly six minutes of compute time.
The results fulfil the needs of the current use case and offer a good basis for further work. In the future, we plan to extend these results in order to reach the required capabilities to offer big data analytics of very large time series data sets directly through HAKOM's TSM system. To achieve this, we are investigating ways to either automatically represent our data in a way that is ready to be analyzed by decentralized computing clusters (like automatic ETL jobs of historic data to Analysis Ready Data formats and analysis friendly file/object storage) or to write custom connectors for Spark to directly target our framework components and handle the data retrieval transparently in the background.

EURO