FINE TUNING APACHE SPARK CONFIGURATIONS USING LARGE LANGUAGE MODELS:

A DATA-DRIVEN APPROACH

Khoa Lang Huynh Dang

University of Information Technology, Vietnam

What?

We introduce a system handling resource orchestration in big data processing systems, in which we have:

- Applied Large Language Models (LLM) to analyze historical data and automatically suggest optimal configurations for Apache Spark.
- Adjusted key parameters such as the number of executors, memory allocation, and partitions to enhance big data processing efficiency.

Why?

- Choosing the optimal Spark configuration is challenging and directly impacts processing time and resource costs.
- Currently, this process relies on data engineers' experience or traditional heuristic-based methods, which have limitations.
- Using LLM automates the process, reduces resource waste, optimizes performance, and improves the accuracy of Spark configuration recommendations.

Overview

Get report from the Spark hosted system

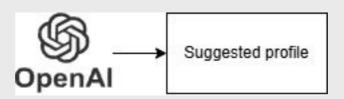


Use LLM for suggestions



Update the system







Description

1. Get report from the Spark hosted system

- Retrieve performance metrics, logs, and historical data from the Spark environment.
- Analyze execution times, resource utilization, and job statuses.
- Identify bottlenecks, inefficiencies, or areas for optimization.

2. Use LLM for suggestions

- Leverage a Large Language Model (LLM) to analyze historical data and predict optimal configurations.
- Generate recommendations for Spark settings, such as executor count, memory allocation, and partition size.
- Compare AI-generated configurations with heuristic-based and manual tuning methods.

3. Update the system

- Apply the recommended Spark configurations to improve performance and resource efficiency.
- Monitor changes in execution speed, cost reduction, and overall system stability.
- Continuously refine the model using feedback from real-world performance data.

