Performance analysis and GPU parameter auto-tuning for SPARK in-memory analytics

As machine learning (ML) and deep learning (DL) are increasingly applied to larger datasets, Apache Spark has become the de facto standard framework for the data pre-processing and feature engineering needed, to prepare raw input data for the learning phase. Leveraging a set of easy-to-use APIs for extract, transform, load (ETL) operations, users are able to process large amounts of data, in a short amount of time, using a farm of servers, either to curate and transform or to analyze and generate insights on data.

While Spark distributes computation across nodes in the form of partitions, within a partition, computation has historically been performed on CPU cores. However, given the parallel nature of many data processing tasks, it’s only natural that the massively parallel architecture of a GPU should be able to parallelize and accelerate data processing queries, in the same way that a GPU accelerates DL in artificial intelligence (AI). Therefore, NVIDIA has worked with the Spark community to implement GPU acceleration through Rapids plugin as part of Spark 3.x. The benefits of acceleration are many. First of all, fewer servers are required in order to perform the same computation, reducing the infrastructure cost. Because of the query acceleration, a reduction in execution time is also expected. Finally, since GPU acceleration is transparent, applications built to run on Spark require no changes in order to reap the benefits of acceleration.

Spark offers a wide variety of configuration parameters which can be adjusted to alter several aspects of its runtime engine for increasing performance. Analyzing and exploring the impact of various configurations on the performance of Spark applications and also examining the inter-correlation between different parameters is a painful procedure for developers, due to i) the high-dimensional configuration space, ii) the huge, cumulative, running time required and iii) the time required to comprehend in depth the purpose of each parameter. To simplify the process of configuration parameter selection, a considerable amount of research works have examined the development of frameworks for automatically tuning Spark applications. In this direction, Nikitopoulou et al. proposed an end-to-end performance auto-tuning framework for Spark in-memory analytics that achieves an average performance gain of ×3.07 for known and ×2.01 for unknown applications, compared to the default configuration.
In this diploma thesis, we aim to extend the aforementioned work for the parameters exposed from Rapids plugin. As a first step, we will examine the provided configuration parameters and identify those that affect performance. Then, we will create a model that will be able to find the configuration that optimizes performance for each application. Finally, we will integrate this model in a Spark cluster, in order to auto-tune all the different submitted applications.

**PREREQUISITES:**
Strong knowledge of Bash, Python
Desirable: Familiarity with Machine Learning, Neural Networks

**SKILLS YOU WILL LEARN:**
In this diploma thesis, you will familiarize with Docker containers, SPARK and Machine Learning techniques.

**RELATED MATERIAL:**
- https://nvidia.github.io/spark-rapids/

**CONTACT INFORMATION:**
Aggelos Ferikoglou Ph.D, NTUA (aferikoglou@microlab.ntua.gr)
Dimosthenis Masouros Ph.D, NTUA (dmasouros@microlab.ntua.gr)
Assistant Professor Sotirios Xydis, HUA (sxydis@hua.gr)
Professor Dimitrios Soudris, NTUA (dsoudris@microlab.ntua.gr)