УДК 004.4
Г. Абоах, Р. Рувімбо, В. Соболь, А. Луцків, канд. техн. наук; доц.
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

РОЗВ'ЯЗАННЯ ЗАДАЧ МАШИННОГО НАВЧАННЯ У СЕРЕДОВИЩАХ ІЗ РОЗПОДІЛЕНОЮ ПАМ'ЯТТЮ

UDC 004.4
H. Aboah, R. Ruwimbo, V. Sobol, A. Lutskiv, Ph.D.; Assoc. Prof.

RESOLVING MACHINE LEARNING TASKS IN DISTRIBUTED MEMORY ENVIRONMENT

Resolving of analytical tasks such as building recommendation or predictive analysis systems involves using of Machine Learning (ML) methods. Usually these methods are implemented in software libraries. The most well-tested ML-methods are implemented in Python libraries. Unfortunately these libraries and solutions can be used only in shared memory environments which are horizontally scale limited. Apache Spark is a distributed memory parallel data processing system which is well horizontally scaled. Other feature of Apache Spark is ability to run locally on a single computer. This feature allows to develop and test ML approaches without having an access to large cluster and also embed Spark into non-distributed applications.

Spark does not offer large amount of ML methods but the most commonly used are implemented in its ml and mllib packages. Among these methods there are different methods to extract features of different types, to classify features, to calculate regression. For ML problems solving in distributed environment Spark offers distributed data types (e.g. DistributedMatrix).

Spark allows to combine resolving of Big Data engineering and Data Science tasks by building pipelines. Spark can be deployed into different environments: physical server or in Kubernetes cloud as a cluster or can be executed as local application on local machine.

Spark has Python API, so data scientist can build solutions by combining Spark Distributed approach with ML-libraries from other tools.

Problems which arise in these solutions related to incompatibility of data formats: usually with Python Pandas library and in Spark RDD, DataFrames and DataSets are used. This autumn Apache Spark developers released version 3.2 [1] which resolves this issue by embedded support of Koalas library (Koalas, the Spark implementation of the popular Pandas library).

Other peculiarities of new Spark version are:

- using Hadoop 3.3.1 libraries (especially performance improvement in AWS S3 object storage support);
- SQL queries using Adaptive Query Execution which improves performance;
- DataSource V2 optimizations related to aggregate pushdown (improvements of operations count, sum, min, max and average);
- Spark Streaming improvement based on using of RocksDB;

and also a few Kubernetes improvements.

Unfortunately not all public cloud providers offer latest version of Apache Spark, so can be used only self-deployed version. For research purposes decided to use helm chart packaged by Bitnami[2] which will be deployed into private Kubernetes cluster.

References.

1. Apache Spark 3.2.0 Documentation. URL: https://spark.apache.org/docs/latest/
2. Apache Spark packaged by Bitnami. URL: https://bitnami.com/stack/spark/helm