UDC 004.04

THE CONCEPT OF DEVELOPING A COMMUNICATION SERVICE FOR STUDENTS, ADMINISTRATION AND POTENTIAL EMPLOYERS

U. BAKLAN, I. BURACHONAK
Polotsk State University, Belarus

This article discusses the development of a communication service for students, administration, and potential employers. A conceptual solution based on data processing and transmission technologies is proposed using the following technologies: Apache Spark, Apache Kafka, Apache Hadoop, PostgreSQL. It describes architectural approaches and solutions that increase the performance, scalability and flexibility of software for the implementation of the presented project.

The modern model of obtaining higher education in our country is collective learning. This is based on the fact that there are more students than teachers. Such a system has its advantages, but the main disadvantage is that it does not provide an opportunity for the student to focus on a specific direction, and as a result, the university graduates a specialist with a profile, covering as wide range of vacancies as possible. However, having received such a specialist, the employer has to spend additional time and money to develop their skills depending on the position. The existing education model can be improved by incorporating an element of personalized learning and automating it. The purpose of automation is to simplify teacher's interaction with a multitude of students while maintaining the element of receiving individual education, which is undoubtedly relevant.

Thus, there is a need for an automated communication system for students, administration, and potential employers, represented by the main functional subsystems (see fig. 1). The system being developed should take upon itself the solution of some of the tasks of the student and the teacher, and subsequently of the potential employer. Moreover, from student's perspective, the system should allow the student to prioritize areas; monitor progress; record achievements in one or another area as a result of which a learning map will be formed and the graduate will be able to find a position that is suitable for the acquired knowledge and skills. When it comes to teachers, the system should provide an opportunity to analyze each student's progress; create individual tasks and plans based on notes; monitor the progress of one student or the whole group, which will allow partial individual training with an acceptable amount of effort and time. The employer, on the other hand, should be able to monitor the students of interest by reviewing the learning maps of a particular person, as well as the ability to adjust the training plan at the later stages of the educational process, so that after graduation the specialist is immediately ready to work in a particular position.

The developed application has no physical load restrictions. The application can generate data and metadata in any amount. It allows you to track user behavior, service, prioritize the selection of subsequent sub-projects and analyze trends that are developed within the service.

There are three main principles of big data processing [1]:

- horizontal scalability (increase in speed and volume due to increased hardware);
- fault tolerance (failure of a certain percentage of servers does not affect the result);
- data locality (the distribution of data to each machine allows you not to lose data in case of server failure).

An example of processing large amounts of data can be ClickStream. Clickstream is a user clicks stream grouped by the user's session ID. Since the amount of data in the first stages is not huge, processing will be carried out on a cluster with a single node (this is any computer connected to the blockchain network) and all clickstream events [2] will be transmitted to the analytical system using a message broker - Apache Kafka.

Apache Kafka is a message manager that is implemented in the Java platform. Kafka has topics in which publishers enter the text of their messages and there are subscribers in threads who read these messages. All messages, without exception, are written to disk during the dispatching process.

The immediate broker message handler is an application written in Python using the Spark framework.

Spark is an Apache project that is positioned as a tool for "lightning-fast cluster computing" [3]. The project is being developed by a thriving free community and is currently the most active of the Apache projects. Spark provides a fast and versatile data processing platform. Compared to Hadoop, Spark accelerates programs more than 100 times in memory, and more than 10 times on-disk [4]. Spark was chosen based on the fact that there is a growing community of developers preferring it, it is provided with support for multiple systems and programming languages and most of the built-in elements. It allows rapid risk assessment, benchmarking, routine checks and analyzes financial and economic indicators.

In our case, the main task of the analytical service is to generate a behavioral "map" of the user, which will subsequently allow analysts and project managers to think through a product development strategy based on the user's behavioral data in a human-readable format.

To reduce the load on the system, it is planned to use Spark Streaming with Batch-mode, which will process streaming data in small batches (batches), which allows to reduce the one-time load and be able to dose the load on the cluster.

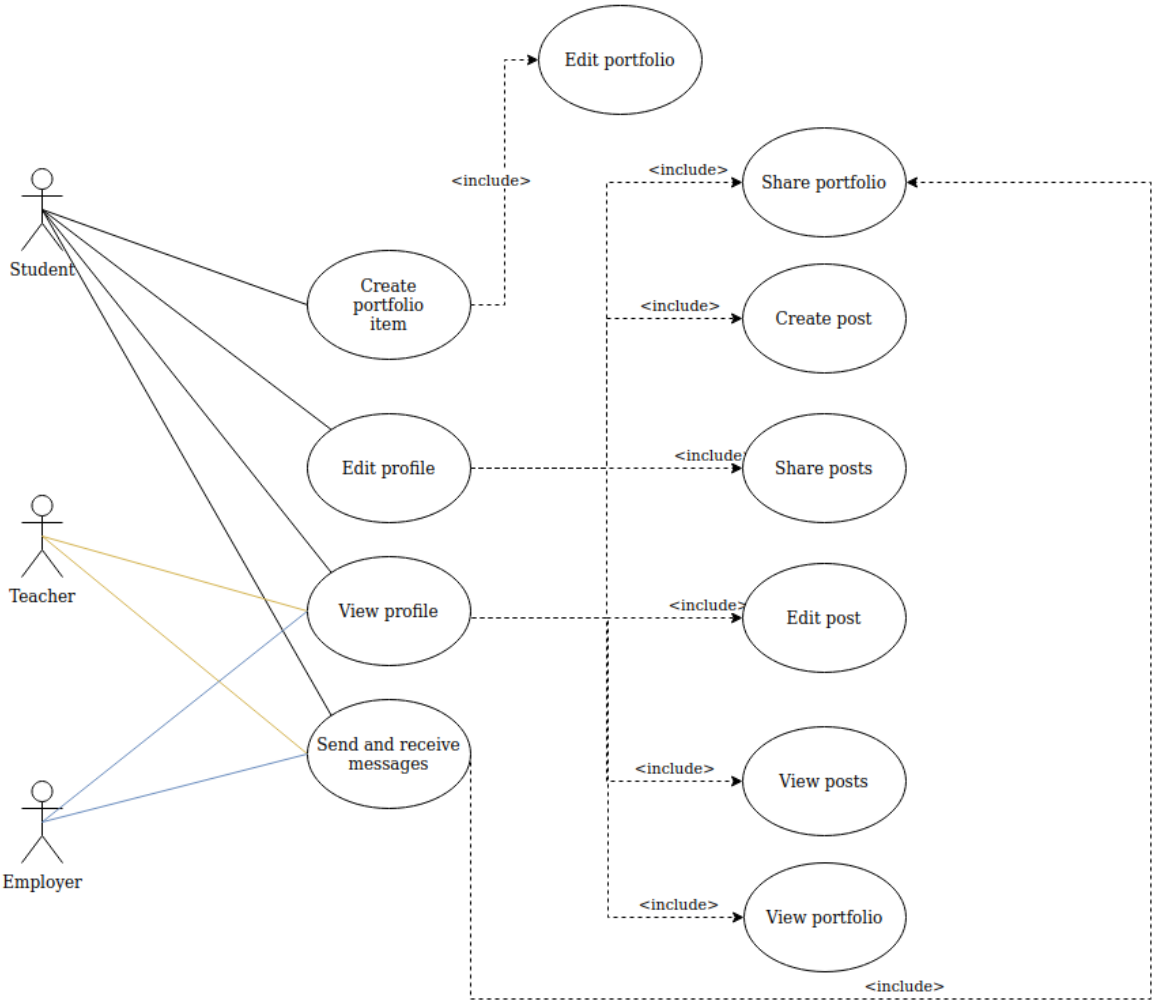

Fig. 1. – Application subsystems

As the application is initially planned to be used by a limited number of users, a minimal configuration with one master node (the node that manages the entire cluster) is sufficient for an analytics cluster. It monitors the other nodes and distributes the load between them) and one worker node (node on which containers are deployed and run), as well as the minimum level of data replication in Kafka. After processing the Spark data, the aggregated data is sent to PostgreSQL for further accessibility from a web application or for subsequent business analysis. The general architecture of data processing in the application is shown in fig. 2.

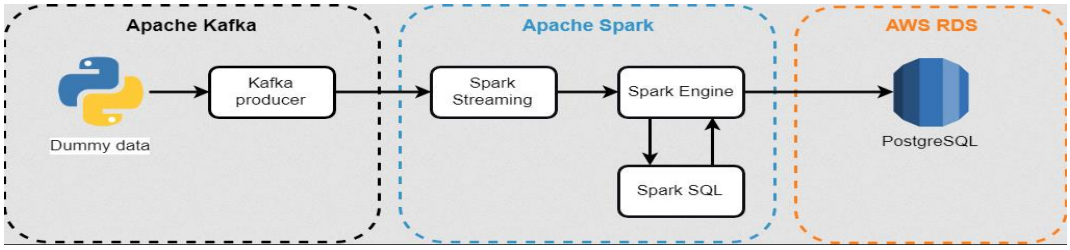

Fig. 2. – Data processing architecture

In this example of analytical processing, data loss is allowed, since a single click or a small amount of it is not critical.

Thus, the use of the presented technology stack will allow the creation of a system that is capable of processing any amounts of data, depending on the requirements and financial constraints. To further increase the amount of data or processing speed, it is enough to expand the cluster, rather than rewrite the existing code. Using the Python language allows writing a code without thinking about the intricacies of the language, which allows to speed up the development process and subsequent refinement.

The developed automated system for communication between students, administration and potential employers will allow:

– to intensify the students' activities in maintaining profiles and of all types of activities performed in the learning process (keeping their progress log allows the student to become flexible in the context of constantly evolving technologies);

– to monitor by the administration of the university the career development for each student during the educational process and the formation of their professional skills and competencies;

– to simplify the process of finding and hiring employees from university graduates for employers;

– and most importantly, based on the collected data (in our case, with the use of the presented technologies, their volume is not limited), to make forecasts of orders for personnel training for certain sectors of the country's national economy and take into account changes in the labor market, avoiding disproportions between supply and demand.

REFERENCES

1. Habr. Habr.com [Electronic Resource] / Big Data, 2021. – Access mode: https://habr.com/ru/post/267361. – Access date 02.03.2021.
2. Clickstream. Clickstream.cc [Electronic Resource] / Clickstream, 2021. Access mode: https://clickstream.cc. – Access date: 02.03.2021.
3. Wikipedia. Apache Spark. [Electronic Resource] / Wikipedia. – Access mode: https://ru.wikipedia.org/wiki/Apache_Spark. – Access date: 03.03.2021.
4. Wikipedia. Apache Hadoop. [Electronic Resource] / Wikipedia. – Access mode: https://ru.wikipedia.org/wiki/Hadoop. – Access date: 06.03.2021.