УДК 004.415.2.031.43
Ілля Федорович, Галина Осухівська, канд. техн. наук, доц.
Тернопільський національний технічний університет імені Івана Пулюя

ВИКОРИСТАННЯ SPARK STREAMING ТА SPARK STRUCTURED STREAMING ДЛЯ ОБРОБКИ ДАНИХ В РЕАЛЬНОМУ ЧАСІ

Illia Fedorovych, Halyna Osukhivska, Ph.D., Assoc. Prof.

USING SPARK STREAMING AND SPARK STRUCTURED STREAMING FOR REAL TIME DATA PROCESSING

У випадку використання Apache Spark для виконання завдань обробки даних в часі, наближеному до реального, постає питання у виборі потокового рушія: Spark Streaming чи Spark Structured Streaming.

Spark Streaming базується на DStream (Discretized Streams – дискретизовані потоки), який представляє собою неперервний ряд RDD (Resilient Distributed Datasets – стійкі розподілені набори даних) [1]. З переваг Spark Streaming можна відзначити підтримку віконного режиму та велику кількість підтримуваних вхідних джерел, а також можливість реалізації власного постачальника даних. Проте, у Spark Streaming виявлено низку проблем, таких як підтримка гарантії обробки «рівно один раз» (exactly-once), обробка надходження даних із запізненням, відмовостійкість, а також невідповідність API потокової обробки (DStream) по відношенню до API RDD та Dataset. Окрім цього, Spark Streaming може здійснювати обробку лише неструктурованих даних, та здійснює потокову обробку не в потоковому режимі, а в мікро-пакетному режимі, що є лише апроксимацією обробки в реальному часі.

Принцип роботи Spark Structured Streaming можна розуміти як необмежену таблицю, яка зростає разом із новими вхідними даними, тобто можна розглядати як обробку потоку, побудовану на Spark SQL [2]. Спрощено кажучи, Structured Streaming внесла кілька нових концепцій у Spark. Гарантія обробки «рівно один раз» – означає, що дані обробляються лише один раз і вихід не містить дублікатів. Час події – однією з помічених проблем із потоковою передачі DStream був порядок обробки, тобто випадок, коли дані, згенеровані раніше, оброблялися після згенерованих пізніше даних. Structured Streaming вирішує цю проблему за допомогою концепції, що називається часом події, яка за деяких умов дозволяє правильно агрегувати запізнілі дані в конвеєрах обробки. Окрім цього, до переваг Spark Structured Streaming можна віднести підтримку обробки структурованих даних та підтримку потокового режиму, який називається Continuous Processing, що дозволяє забезпечити справжню обробку даних в реальному часі (з періодом затримки ~1 мс) та гарантією доставки «щонайменше один раз» (at-least-once). Continuous Processing поки що знаходиться на експериментальному етапі, та окрім спеціальних джерел для налагодження, підтримує лише Apache Kafka в якості джерел для входу та виходу даних [3].

Література

1. Spark Streaming [Електронний ресурс] – Режим доступу до ресурсу: https://spark.apache.org/docs/latest/streaming-programming-guide.html.

2. Structured streaming: A declarative API for real-time applications in Apache Spark / [M. Armbrust, T. Das, J. Torres та ін.]. // Proc. Int. Conf. Manage. Data. – 2018. – С. 601–613.

3. Structured Streaming Programming Guide [Електронний ресурс] – Режим доступу до ресурсу: https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html.