УДК 004.048:004.6:004.8

Т.Д. Польгуль

ПОРІВНЯЛЬНИЙ АНАЛІЗ APACHE SPARK ТА APACHE FLINK ДЛЯ РОБОТИ З BIG DATA

Вінницький національний технічний університет

Анотація

У даній роботі здійснено порівняльний аналіз програмних платформ оброблення даних Apache Spark та Apache Flink для роботи з Big Data.

Ключові слова: Apache Spark, Apache Flink, Big Data, обробка даних.

Abstract

A comparative analysis of data processing software platforms Apache Spark and Apache Flink for working with Big Data.

Keywords: Apache Spark, Apache Flink, Big Data, Data Processing.

Метою даного дослідження є порівняльний аналіз програмних платформ оброблення даних Apache Flink та Apache Spark в контексті забезпечення кращої продуктивності при роботі з Big Data.

Характеризуючи спільні риси програмних засобів для роботи з Big Data, потрібно зазначити, що Flink і Spark - це універсальні платформи оброблення даних і проекти верхнього рівня Apache Software Foundation (ASF). Вони мають широку сферу застосування і можуть використовуватися для десятків сценаріїв роботи з Big Data. Завдяки розширенням, таким як [1-4]:

- запити SQL (Spark: Spark SQL; Flink: MRQL);
- обробка Graph (Spark: Graphx; Flink: Spargel (базовий) і Gelly (бібліотека));
- машинне навчання (Spark: MLlib; Flink: Flink ML);
- потокова обробка (Spark Streaming; Flink Streaming).

Вказані платформи оброблення даних здатні працювати в автономному режимі, але також є можливість їх використання поверх Hadoop (YARN – Yet Another Resource Negotiator, HDFS – Hadoop Distributed File System). Також, вони забезпечують відносно ефективну роботу з Big Data, проте, спосіб, яким вони досягають ефекту і задачі, на яких вони спеціалізуються, розрізняються [1].

Flink оптимізований для циклічних або ітераційних процесів за допомогою ітераційних перетворень в колекціях. Це досягається за рахунок оптимізації об'єднання алгоритмів, оператора об’єднання і повторного використання розбиття і сортування. Однак Flink - це також потужний інструмент для пакетної обробки. Flink Streaming обробляють потоки даних як справжні потоки, тобто елементи даних негайно "конвертуються" через програму потокової передачі, як тільки вони прибувають. Це дозволяє виконувати гнучкі window операції на потоках. Крім цього, Flink забезпечує достатньо високу сумісність, що надає можливості використовувати існуючий storm-, map reduce- код тощо на Flink [1-4].

Можна виділити переваги Flink за такими критеріями:

- набори даних: надає здатність обробляти набори даних більші, ніж RAM;

- потік обробки: Apache Flink - це система потокової обробки, яка може обробляти рядок за рядком в реальному часі;

- ітерації: експлуатуючи свою потокову архітектуру, Flink містить вбудовані ітеративні оператори;

- управління пам'яттю: у Flink-а автоматичне управління пам'яттю;

- потік даних: на відміну від процедурної парадигми програмування, Flink наслідує підхід розподілених потоків даних. Для набору даних операцій, де проміжні результати

необхідно враховувати разом з регулярним входом операції, широкомовні змінні використовуються для розподілу наперед розрахованих результатів до всіх робочих вузлів.

До недоліку Flink можна віднести [1-3]:

- зрілість: Flink все ще перебуває на ранній стадії розвитку і має невелику кількість виробництв розгортання.

Spark з іншого боку, грунтується на щільно розподілених наборах даних (RDDs). В основному in-memory структури даних надають переваги функціональним парадигмам програмування Spark. Платформа Spark здатна здійснювати обчислення великих пакетів, закріпивши їх у пам'яті. Spark Streaming перетворюють потоки даних в міні-пакети, тобто, збирають всі дані, що прибувають протягом певного періоду часу і запускають регулярну пакетну програму на зібраних даних. У той час як пакетна програма запущена, збираються дані для наступної міні-партії [1, 4].

Можна виділити такі переваги (хоча, в окремих випадках це може виявитись і недоліком) за такими критеріями [1, 4, 5]:

- потік обробки: це пакетно-орієнтована система, яка працює з чанками (пакетами) даних, що називаються RDDs;

- ітерації: Spark також підтримує перебір даних, але лише при роботі з пакетами.

- управління пам'яттю: Spark jobs повинні бути оптимізовані і адаптовані до конкретних наборів даних, тому що потрібно вручну управляти розділенням і кешуванням, якщо необхідно отримати ці права;

- зрілість: Spark є достатньо зрілим проектом.

До недоліку Spark необхідно віднести також те, що на відміну від Flink, Spark не здатний обробляти набори даних більші, ніж RAM до версії 1.5.x [1], а також недоліки роботи з мікропакетами у Spark [5].

Результати порівняння програмних платформ оброблення даних Apache Spark та Apache Flink наведено в таблиці 1.

Таблиця 1

Результати порівняння Apache Spark та Apache Flink

	Spark Streaming	**Flink**
Оперування даними	Відсутнє дублювання	
Пропускна здатність	Висока	
Накладні витрати відмовостійкості	Низькі	
Обчислювальна модель	Мікропакети	Потоки
Window критерії	Засновані на часі	Засновані на кількості записів або визначені користувачем
Управління пам’яттю	Налаштовуване (необхідно конфігурувати)	Автоматичне

Таким чином, Apache Flink забезпечує кращу продуктивність, ніж Apache Spark за рахунок [1]:

- менеджера специфічної пам’яті,;

- нативних ітераційних операторів, які приводять до значного пришвидшення роботи програм машинного навчання та обробки графів,

- вбудованого автоматичного оптимізатора;

- конвеєрної обробки даних, яка у Flink є більш ефективною, ніж у Spark.

Незважаючи на всі переваги обох програмних платформ оброблення даних, натепер ні Flink, ні Spark не можуть повністю замінити Hadoop. Проте Flink є заміною для Hadoop MapReduce, який працює в обох режимах: командному і потоковому, усуваючи такі jobs як map і reduce на користь орієнтованого графа, який використовує зберігання в пам'яті для масового

збільшення продуктивності. Але HDFS (Hadoop Distributed File System) і YARN (Yet Another Resource Negotiator), які є частиною великої екосистеми Hadoop не можуть бути заміненими Flink-ом [1-3].

Отже, в проведених наукових дослідженнях здійснено порівняльний аналіз програмних платформ оброблення даних Apache Spark та Apache Flink, в контексті роботи з Big Data. Відзначено особливості, за рахунок яких Apache Flink забезпечує кращу продуктивність, ніж Apache Spark.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flink vs. Spark [Електронний ресурс]. – Режим доступу: http://www.slideshare.net/sbaltagi/flink-vs-spark
2. Apache Flink [Електронний ресурс]. – Режим доступу: https://flink.apache.org/
3. Flink Streaming [Електронний ресурс]. – Режим доступу: https://flink.apache.org/features.html
4. Spark [Електронний ресурс]. – Режим доступу: http://spark.apache.org/
5. Distributed Stream and Graph Processing with Apache Flink [Електронний ресурс]. – Режим доступу: http://www.meetup.com/Bay-Area-Apache-Flink-Meetup/events/224189524/

Польгуль Тетяна Дмитрівна **–** аспірант кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: tanapolg93@gmail.com

Науковий керівник: Яровий Андрій Анатолійович **–** д.т.н., професор кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця

Tetiana D. Polhul **–** postgraduate student of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: tanapolg93@gmail.com

Scientific Supervisor: Andriy A. Yarovyy **–** Doctor Sc. (Eng), Professor, Professor of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia