Технічні науки

УДК 004.424

Бєлєвцов Віталій Вікторович

студент

Харківського національного університету радіоелектроніки

Белевцов Виталий Викторович

студент

Харьковского национального университета радиоэлектроники

Bielievtsov Vitalii

Student of the

Kharkiv National University of Radioelectronics

Тукало Ростислав Андрійович

студент

Харківського національного університету радіоелектроніки

Тукало Ростислав Андреевич

студент

Харьковского национального университета радиоэлектроники

Tukalo Rostyslav

Student of the

Kharkiv National University of Radioelectronics

ВИКОРИСТАННЯ ПЛАТФОРМИ .NET ДЛЯ РОБОТИ З ВЕЛИКИМИ ДАНИМИ
ИСПОЛЬЗОВАНИЕ ПЛАТФОРМЫ .NET ДЛЯ РАБОТЫ С БОЛЬШИМИ ДАННЫМИ
USING THE .NET PLATFORM FOR BIG DATA PROCESSING

Анотація. *Досліджено можливості використання платформи .NET у якості інструменту для роботи з великими даними.*

Ключові слова: *великі дані, ефективність, обчислення, платформа .NET, програмна інженерія.*

Аннотация. *Исследованы возможности использования платформы .NET в качестве инструмента для работы с большими данными.*

Ключевые слова: *большие данные, вычисления, платформа .NET, программная инженерия, эффективность.*

Summary. *Possibility of using the .NET platform as an instrument for big data processing has been investigated.*

Key words: *big data, computing, efficiency, .NET platform, software engineering.*

Із розвитком комп'ютерних технологій проблема ефективної обробки та інтелектуального аналізу великих масивів даних посідає важливе місце серед інших у цій області. Під поняттям великих даних розуміється саме англомовне big data. Основне та найпоширеніше трактування визначає цей термін як набори структурованої або неструктурованої інформації, настільки великі, що класичні методи обробки та вивчення даних не відповідають обсягам цих масивів. Також існує наступне визначення: феноменальне прискорення темпів збільшення кількості даних та їх ускладнення (до прикладу, вважається, що обсяг інформації в світі зростає щороку на 30%, а за п'ять останніх років людство накопичило більше знань та даних, аніж за усю свою попередню історію). Очевидно, що саме перед галуззю інформаційних технологій постає питання осягнення та охоплення всієї маси інформації, що з часом зростає експоненціально, а отже виклики майбутнього потребують

сучасних, потужних та ефективних технічних рішень для отримання знань у майже будь-якій царині життя людства.

У big data спільноті, очевидно, вже існує декілька основних програмних інструментів, що поширені серед спеціалістів з обробки даних та є усталеними для використання. Зокрема, це мови програмування Python, R та Scala. Їх об'єднує те, що ці мови є мультипарадигмальними (поєднують функціональне та об'єктноорієнтоване програмування), а для виконання програм код інтерпретується (хоча Scala надає можливість використовувати і компілятор, і інтерпретатор). Використання саме цих мов зумовлене їх високою продуктивністю, а також наявністю великої кількості доповнень та бібліотек для потужної, зручної та гнучкої обробки інформації.

Ознайомимося ж нині з платформою .NET. Однією з основних ідей цієї програмної технології є можливість взаємодії коду та служб, що написані різними мовами програмування – цей аспект забезпечується за допомогою загальномовного середовища виконання (Common Language Runtime, CLR), що виконує на даному процесорі байткод CIL або MSIL, у який, у свою чергу, перетворюються всі .NET-сумісні мови (найбільш поширені – C#, F#, VB.NET). Інші ж компоненти платформи належать до інструментарію розробки – серед іншого, це технології доступу до даних, бібліотеки класів, засоби паралельного виконання. .NET є кросплатформовою технологією, що дозволяє виконувати програми на інших операційних системах, окрім Windows.

До основних галузей застосування платформи .NET належать розробка серверних застосувань, API, розробка ігрових застосувань (у тому числі за домопогою рушія Unity) та розробка мобільних застосувань (за допомогою технології Xamarin, що розширює .NET бібліотеками класів та інструментами для розробки на Android та iOS). Проте віднедавна список

можливостей використання .NET поповнилися ще однією галуззю – обробка великих даних, на яку і звертаємо увагу у даній статті.

У жовтні 2020 року було анонсовано вихід першої стабільної версії фреймворку .NET for Apache Spark. Apache Spark – це, в свою чергу, також big data фреймворк, що складається з ядра, інструмента для аналітичної обробки даних за допомогою SQL-запитів, модуля розподіленої обробки графів та інших підсистем. Ця технологія, по суті, надає API-інтерфейси для різних мов програмування, для яких можливо створювати власні фреймворки для використання обчислювальних та аналітичних можливостей Apache Spark. На момент впровадження підтримки .NET серед основних підтримуваних мов були R, Python, Scala та Java.

Передумовою створення фреймворку для взаємодії .NET та Apache Spark став широкий запит .NET-спільноти на більш простий спосіб розробки big data застосунків саме на платформі .NET, щоб уникнути необхідності вивчення Scala або Python для цієї мети. Відтак було створено команду для написання проєкту з відкритим кодом. Найперша публічна версія була представлена в квітні 2019 року, з того часу до релізу першої стабільної версії відбулося 12 передвипусків. Завдяки гнучкості та модульності платформи .NET розробники мають можливість доповнювати кодом big data систем інші частини своїх проєктів та застосунків, а також безперешкодно використовувати свої знання, напрацювання, код, а також взаємодіяти з будь-якими бібліотеками, що вже існують; у якості прикладу можна навести бібліотеку для машинного навчання ML.NET, що у поєднанні зі Spark може стати потужним конкурентом для інших усталених способів та програмних систем для розробки високоінтелектуальних застосунків.

Розробники заявляють, що .NET for Apache Spark є швидшим, аніж конкуренти для мов Python та Scala, а в деяких випадках фреймворк PySpark для Python програє в продуктивності у два рази. Спробуємо ж

провести власну оцінку потенціалу фреймворків для .NET та Python. Для цього використовуватимемо один із модулів, що надається Apache Spark, а саме інструмент для аналітичної обробки за допомогою SQL-запитів. Виконуватимемо задачу з підрахунку кількості слів у заданому текстовому файлі. У нашому випадку використовуватимемо текст промови В. Черчілля (https://en.wikisource.org/wiki/We_shall_fight_on_the_beaches), повторений у файлі 10000 разів (результуючий txt-файл містить 37920000 слів, обсяг – 200 мегабайт) без розділових знаків.

В обох програмах алгоритм наступний: розпочати сесію Spark, тобто під'єднати код програми через фреймворк до основних інструментів та потужностей на комп'ютері, почати вимір часу, завантажити текст із тестового файлу, підрахувати за допомогою SQL-інструменту кількість слів, вивести результат (20 найчастіших слів) на консоль, зупинити вимір часу та вивести результат виміру, припинити Spark-сесію. Отже, кількість та зміст кроків у програмах, не враховуючи особливостей кожної з мов, абсолютно ідентичні.

Наведемо код методу, що виконує підрахунок, на C#:

```
// Створюємо сесію Spark
SparkSession spark =
     SparkSession
          .Builder()
          .AppName("word_count_sample")
          .GetOrCreate();

// Починаємо вимірювати час
Stopwatch stopWatch = new Stopwatch();
stopWatch.Start();

// Створюємо об'єкт DataFrame
DataFrame dataFrame = spark.Read().Text("./Churchill.txt");
```

```
// Підраховуємо результат
DataFrame words =
     dataFrame
          .Select(Split(Col("value"), " ").Alias("words"))
          .Select(Explode(Col("words")).Alias("word"))
          .GroupBy("word")
          .Count()
          .OrderBy(Col("count").Desc());

// Виводимо результат
words.Show();

// Виводимо виміряний час
stopWatch.Stop();
TimeSpan ts = stopWatch.Elapsed;
Console.WriteLine("RunTime " + ts);

// Припиняємо сесію
spark.Stop();
```

Так само наведемо код програми на Python. Оскільки опосередковано використовується один і той самий фреймворк, сигнатури використовуваних функцій часто схожі і можна простежити однаковість виконуваних дій.

```
spark = SparkSession \
    .builder \
    .appName("word_count_sample") \
    .getOrCreate()

start = time.time()

words = spark.read.text("./Churchill.txt")
```

```
words.withColumn('words', split(col('value'), ' ')) \
    .withColumn('word', explode(col('words'))) \
    .groupBy('word') \
    .agg(count('word').alias('count')) \
    .orderBy('count', ascending=False) \
    .show(20)

print("RunTime", time.time() - start)

spark.stop()
```

Після запуску обох програм (очевидно, на одній машині для більшої точності порівняння, проте для мови C# використали режим Release для оптимізації роботи програми, не використовуючи зайві інструменти для дебагінгу) отримали наступні результати. Програми отримали однакові результати, але час виконання трохи відрізняється на користь мови Python – платформа .NET виконувала цю задачу в середньому на 12-20% повільніше.

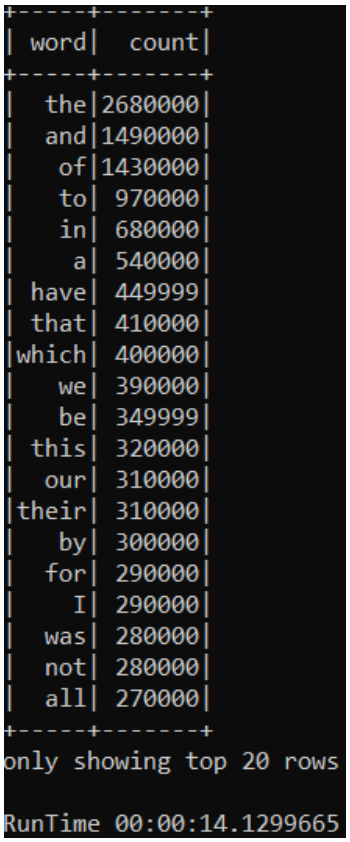

Рис. 1. Результат підрахунку кількості слів та виміру часу роботи програми для платформи .NET

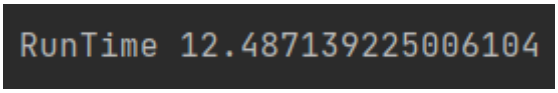

Рис. 2. Результат виміру часу роботи програми для мови Python

Отже, незважаючи на заяви розробників, наш своєрідний бенчмарк виявив невелику перевагу мови Python над платформою .NET і мовою програмування в нашому випадку C#. Об'єктивність, порівняно з професійними системами оцінки продуктивності платформ та фреймворків, звичайно, нижча, але, так чи інакше, можемо заявити про наступне: перспектива використання .NET у якості основного інструменту в руках big data аналітика є цілком реальною, а тим паче у зв'язці з іншими засобами розробки застосунків, як-от з бібліотеками для машинного навчання, що вже існують. Необхідно звернути увагу на певну різницю в часовій ефективності різних мов в певних ситуаціях використання, проте

цей аспект можна виправдати тим, що можливо вирішувати задачі big data за участі .NET-спеціалістів без необхідності для них вивчати Python чи Scala або наймати нових працівників з такими навичками.

Література

1. .NET for Apache Spark | Big data analytics. 2021. URL: https://dotnet.microsoft.com/apps/data/spark
2. Что такое Apache Spark: история, архитектура, особенности работы. 2019. URL: https://www.bigdataschool.ru/wiki/spark
3. Microsoft and the .NET Foundation announce the release of version 1.0 of .NET for Apache Spark. 2020. URL: https://techcommunity.microsoft.com/t5/azure-synapse-analytics/microsoft-and-the-net-foundation-announce-the-release-of-version/ba-p/1820908