УДК 004.6 DOI: https://doi.org/10.30837/IYF.IIS.2024.632

ДОСЛІДЖЕННЯ ВПЛИВУ НАЛАШТУВАНЬ КОНФІГУРАЦІЙНИХ ПАРАМЕТРІВ НА ПРОДУКТИВНІСТЬ РЕЖИМІВ РОЗГОРТАННЯ APACHE SPARK

Коптілов Н. С.
Науковий керівник – д.т.н., проф. Мінухін С. В.
Харківський національний економічний університет ім. С. Кузнеця
м. Харків, Україна
e-mail: koptilov.nikita.s@hneu.net

The study is devoted to a comparative analysis of the application of different modes of using Apache Spark for the Word Count test dataset. An approach is proposed in which, based on the determination of the main configuration parameters of the Standalone and Hadoop Yarn modes and taking into account the limitations on the local deployment resource. A choice of configuration parameters and their changes in accordance with the default settings is proposed. The results are shown to substantiate its effectiveness based on the obtained quantitative estimates of the execution time.

Великі дані – це не просто технологічний прорив, а й значний тренд змін у сучасному світі. У світі використання технологій великих даних однією з провідних є Apache Spark [1]. Він дозволяє ефективно обробляти та аналізувати великі обсяги даних, роблячи цей процес більш доступним і швидким. Apache Spark – це універсальна високопродуктивна обчислювальна платформа, яка використовує компоненти Apache Hadoop HDFS та MapReduce [2] для виконання різних завдань великомасштабної обробки даних.

Особливістю даного дослідження є аналіз впливу налаштуваннь конфігураційних параметрів кластера Apache Spark за умов обмеженості обчислювальних ресурсів, що й обумовлює можливість використання результатів моделювання роботи кластеру для різних об’ємів вхідних файлів (робіт) для створення подальших сценаріїв покращення продуктивності фреймворку [3].

Для проведення експериментів було застосувано локальний ресурс з хостовою ОС Windows 10 та з використанням віртуального середовища VirtualBox з гостьвою ОС Ubuntu Xenial 16.04. Кластер Apache Spark мав архітектуру із 3 вузлами – 1 майстер-вузлом (master) та 2 робочими вузлами (workers). Для порівняльного аналізу використовувалися режими розгортання Apache Spark Standalone та Apache Hadoop Yarn у режимі FIFO (з двома воркерами та одним майстром).

Згідно з результатами роботи [4], було обрано основні параметри для проведення дослідження.

У режимі Apache Spark Standalone було налаштувано такі параметри (файл конфігурації Spark spark-defaults.conf):

− пам'ять: spark.executor.memory 2048 МБ – об’єм пам'яті, який виділяється для кожного виконавця (executor); spark.driver.memory 1 ГБ – обєм пам'яті, який виділяється для драйвера Spark;

− кількість виконавців (executors) та ядер: spark.executor.instances 2; кількість ядер на 1 виконавця – 1;

− рівень паралелізму (parallelism) – spark.default.parallelism 4-32 : визначає кількість розділів у RDD за замовчуванням, які повертаються такими перетвореннями, як join, reduceByKey та parallelize.

Інші налаштування: spark.sql.shuffle.partitions 4 – кількість роздільних потоків, які використовуються для операцій перемішування в Apache Spark.

Режим Apache Hadoop YARN FIFO використовується для оброблення файлів невеликих обємів з підходом до розподілу ресурсів, в якому завдання виконуються в порядку їхнього надходження. Для цього режиму було обрано такі налаштування режиму (файл конфігурації YARN yarn-site.xml):

− пам'ять для контейнерів: yarn.scheduler.minimum-allocation-mb – 256 МБ – мінімальний розмір пам'яті (МБ), який може бути призначений для контейнера; yarn.scheduler.maximum-allocation-mb – 4096 МБ – максимальний обєм пам'яті (МБ), який може бути призначений для контейнера;

− кількість ресурсів: yarn.nodemanager.resource.memory-mb – 4 796 МБ – загальний обсяг пам'яті (МБ), доступний на кожному вузлі кластера для запуску контейнерів. yarn.scheduler.maximum-allocation-vcores – 4 – максимальна кількість віртуальних ядер для кожного контейнера.

У системі Hadoop YARN контейнери використовуються для запуску завдань обробки даних, таких як MapReduce або додатків Apache Spark. Віртуальні ядра (vcores) є абстракцією, що представляє кількість потоків, які може використовувати контейнер для паралельного виконання завдань. У дослідженні запропоновано методологічний підхід до ефективного налаштування парметрів Apache Spark, який полягає в тому, що в ньому поетапно враховуються можливі загальні динамічні зміни у налаштуваннях з використанням ієрархії найбільш впливових параметрів та рекомендовані зміни налаштувань з врахуванням обмеженості наявних обчислювальних ресурсів системи.

Дослідження продуктивності наведених режимів розгортання проведено на наборі тестових даних WordCount [3, 4], який використовується для текстових даних великих обсягів для пошукових систем, соціальних медіа, аналітики новинних порталів та інш., а також з врахуванням результатів роботи [5]. Для проведення експериментів було розроблено скрипт на мові Python, який дозволяє створювати синтезовані текстові файли різних розмірів, що структуровані за кількістю рядків (абзаців).

Для проведення експериментів було створено тестові текстові файли Word Count різних розмірів – від 0.5 ГБ до 2.5 ГБ з кроком 0.5 ГБ.

Запропоновано наступні зміни параметрів налаштувань цих режимів.

Для Apache Spark Standalone:

spark_worker_memory="3G",spark_worker_cores="4",spark_executor_instances="2", spark_driver_memory="1G", spark.default.parallelism – 16.

Для Apache Hadoop Yarn:

yarn.nodemanager.resource.memory-mb 4796, yarn.scheduler.maximum-allocation-mb 4096, yarn.nodemanager.resource.cpu-vcores 4, yarn.scheduler.maximum-allocation-vcores 4, mapreduce.job.maps 2, mapreduce.job.reduces 2, yarn.app.mapreduce.am.resource.mb 4096.

Статистичний аналіз результатів, проведений на основі обчислень для датасетів з оптимізованими конфігураціями параметрів режимів Standalone та Yarn, показав зменшення часу виконання за конфігурацією Yarn: для файлів розміром 0,5 ГБ, 1 ГБ та 1,5 ГБ – на 15-30%, для файлів розміром 2 ГБ та 2,5 ГБ – на 10-15%.

Таким чином, порівняльний аналіз використання режимів Apache Spark Standalone та Apache Hadoop YARN FIFO для виконання тестових завдань Word Count довів, що Apache Hadoop YARN FIFO продемонстрував кращі результати за запропонованими змінами значень параметрів налаштувань. Доведено, що ефективність режиму розгортання Apache Spark YARN FIFO визначається потребами щодо ресурсів, розмірами вхідних файлів та врахуванням характеристик наявних обчислювальних ресурсів системи. Це значним чином впливає на вибір складу конфігураційних параметрів фреймворків, зокрема, в умовах обмеженості потужностей обчислювальних ресурсів та при використанні віртуальних середовищ.

Список використаних джерел:

1. Zaharia M. et al. Apache Spark: A Unified engine for large-scale data analytics. // Communications of the ACM. 2016. Vol. 59. Issue 11. P. 56–65. https://doi.org/10.1145/2934664.

2. Apache Hadoop. URL: https://hadoop.apache.org/ (дата звернення: 10.02.2024).

3. Priyanka R. Z., Kumar P. N. V. S. Pavan. Comparing the Word Count Execution Time in Hadoop & Spark. // International Journal of Innovative Science, Engineering & Technology. 2016. Vol. 3. Issue 10. P. 25–34.

4. Benlachimi Y., Yazidi A. El, Hasnaoui M. L. A Comparative Analysis of Hadoop and Spark Frameworks using Word Count Algorithm. // International Journal of Advanced Computer Science and Applications. 2021. Vol. 12. No 4. P. 778–788. DOI:10.14569/IJACSA.2021.0120495.

5. Мінухін С. В. Дослідження продуктивності кластера Apache Spark на платформі Azure для методів машинного навчання // Збірник наукових праць Харківського національного університету Повітряних Сил. 2020. №. 1 (63). С. 81–88.