1. Введение

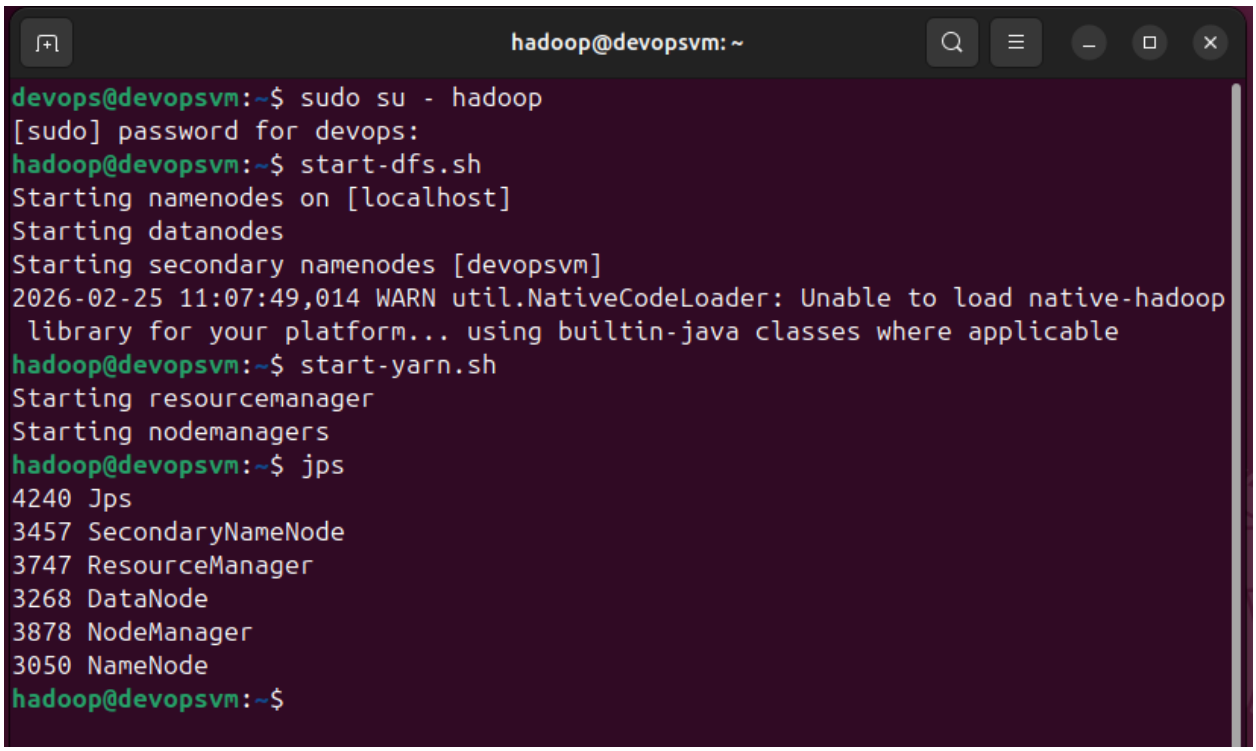
Цель — Изучение принципов работы с распределенной файловой системой HDFS и фреймворком Apache Spark для обработки больших данных. В ходе работы необходимо освоить загрузку данных в HDFS, выполнить очистку и предобработку данных с использованием PySpark (RDD и DataFrame), применить Spark SQL для аналитических запросов и визуализировать полученные результаты для поддержки принятия управленческих решений.

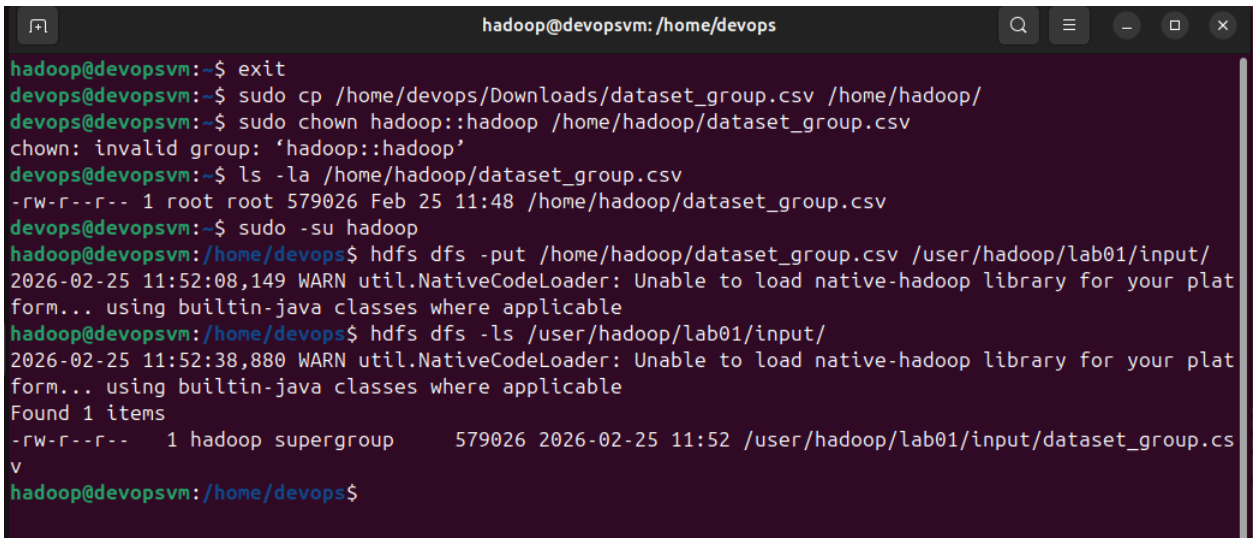
Постановка бизнес-задачи: необходимо проанализировать данные о пользовательских корзинах для выявления ключевых паттернов потребления и брошенных корзин. Требуется определить топ-5 товаров, которые чаще всего остаются в брошенных корзинах, чтобы разработать стратегии по удержанию клиентов. Также необходимо выявить пары товаров, которые покупатели чаще всего приобретают вместе для оптимизации расположения товаров в магазине.

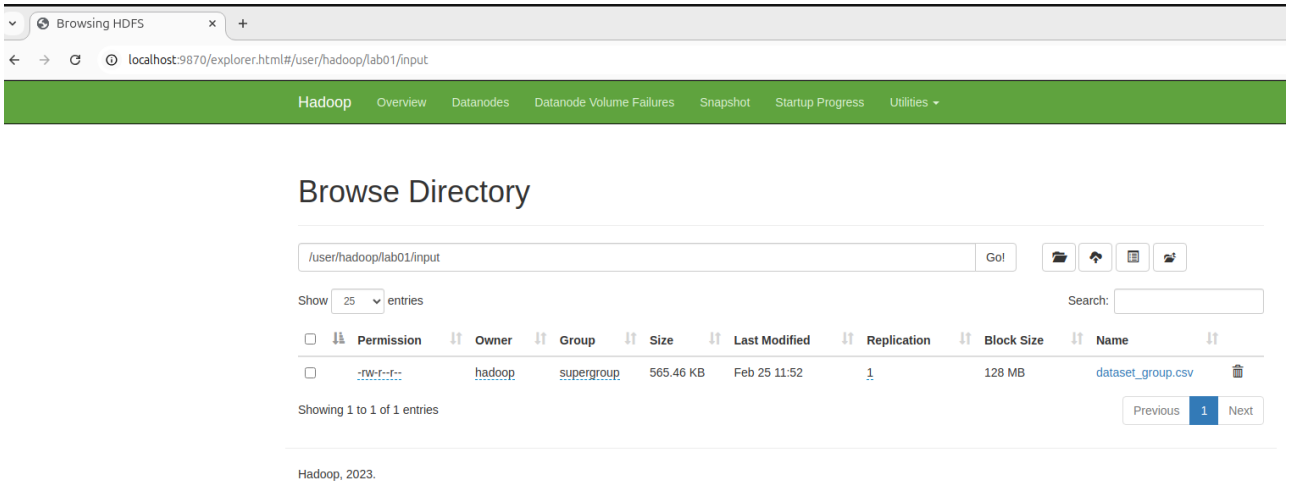
Описание данных:

1) date — дата добавления в корзину
2) basket_number — номер корзины
3) product — товар
4) abandoned — статус брошенной корзины

2. Ход работы


hadoop@devopsvm: ~
devops@devopsvm:~$ sudo su - hadoop
[sudo] password for devops:
hadoop@devopsvm:~$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [devopsvm]
2026-02-25 11:07:49,014 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@devopsvm:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@devopsvm:~$ jps
4240 Jps
3457 SecondaryNameNode
3747 ResourceManager
3268 DataNode
3878 NodeManager
3050 NameNode
hadoop@devopsvm:~$

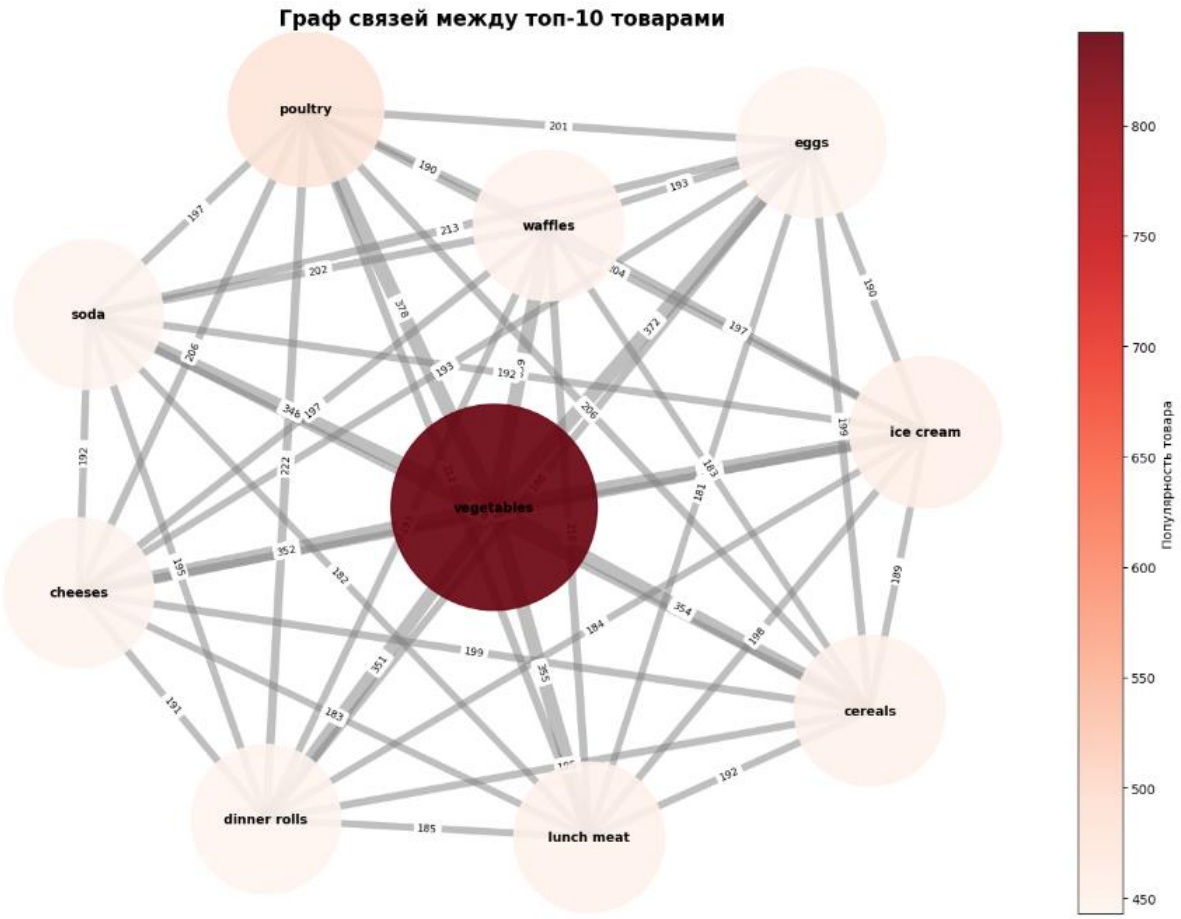

hadoop@devopsvm: /home/devops
hadoop@devopsvm:~$ exit
devops@devopsvm:~$ sudo cp /home/devops/Downloads/dataset_group.csv /home/hadoop/
devops@devopsvm:~$ sudo chown hadoop::hadoop /home/hadoop/dataset_group.csv
chown: invalid group: 'hadoop::hadoop'
devops@devopsvm:~$ ls -la /home/hadoop/dataset_group.csv
-rw-r--r-- 1 root root 579026 Feb 25 11:48 /home/hadoop/dataset_group.csv
devops@devopsvm:~$ sudo -su hadoop
hadoop@devopsvm:/home/devops$ hdfs dfs -put /home/hadoop/dataset_group.csv /user/hadoop/lab01/input/
2026-02-25 11:52:08,149 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@devopsvm:/home/devops$ hdfs dfs -ls /user/hadoop/lab01/input/
2026-02-25 11:52:38,880 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 hadoop supergroup   579026 2026-02-25 11:52 /user/hadoop/lab01/input/dataset_group.csv
hadoop@devopsvm:/home/devops$


Browsing HDFS
localhost:9870/explorer.html#/user/hadoop/lab01/input
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/user/hadoop/lab01/input
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
hadoop
supergroup
565.46 KB
Feb 25 11:52
1
128 MB
dataset_group.csv
Showing 1 to 1 of 1 entries
Previous
1
Next
Hadoop, 2023.

3. Анализ

```
# SQL запрос для нахождения пар товаров в одной корзине
pairs_sql = spark.sql("""
    WITH basket_products AS (
        SELECT basket_number, collect_list(product) as products
        FROM purchases
        GROUP BY basket_number
        HAVING size(products) >= 2
    ),
    numbered_products AS (
        SELECT
            basket_number,
            product,
            ROW_NUMBER() OVER (PARTITION BY basket_number ORDER BY product) as rn
        FROM purchases
    ),
    product_pairs AS (
        SELECT
            a.basket_number,
            a.product as product1,
            b.product as product2
        FROM numbered_products a
        JOIN numbered_products b ON a.basket_number = b.basket_number AND a.rn < b.rn
    )
    SELECT
        product1,
        product2,
        COUNT(*) as co_occurrence
    FROM product_pairs
    GROUP BY product1, product2
    ORDER BY co_occurrence DESC
    LIMIT 10
""")
```

```
Топ-10 пар товаров, которые чаще всего покупают вместе:
+----------------+----------+-------------+
|        product1|  product2|co_occurrence|
+----------------+----------+-------------+
|         poultry|vegetables|          378|
|            eggs|vegetables|          372|
|      vegetables|    yogurt|          363|
|      vegetables|   waffles|          359|
|      lunch meat|vegetables|          355|
|   aluminum foil|vegetables|          354|
|         cereals|vegetables|          354|
|         cheeses|vegetables|          352|
|laundry detergent|vegetables|          352|
|    dinner rolls|vegetables|          351|
+----------------+----------+-------------+
```



Элементы графа и их бизнес-смысл:

- **Узлы** — показывают топ-10 самых популярных товаров в ассортименте
- **Размер узла** — отражает популярность товара (частоту покупок)
- **Ребра** — показывают, что товары часто покупают вместе
- **Толщина ребра** — отражает силу связи (как часто пару покупают вместе)
- **Цвет узла** — дополнительная визуализация популярности (от светлого к темному)

Самым популярным товаром в ассортименте являются овощи (vegetables). Это ожидаемо, так как овощи относятся к категории товаров повседневного спроса и присутствуют практически в каждой потребительской корзине независимо от типа покупки. Высокая популярность овощей делает их ключевым звеном, вокруг которого формируются покупательские предпочтения, что подтверждается многочисленными связями этого товара с другими позициями в построенном графе. Данный факт имеет важное бизнес-

значение: овощи должны всегда присутствовать в достаточном количестве и высоком качестве, поскольку их отсутствие может привести к потере не только самой покупки, но и связанных с ней товаров.

4. Выводы

В ходе выполнения данной работы были успешно применены распределенная файловая система HDFS и фреймворк Apache Spark для анализа пользовательских корзин, что позволило в полной мере оценить их преимущества при работе с большими объемами информации. Использование HDFS обеспечило надежное хранение исходных данных и возможность их масштабирования без потери производительности, а Spark продемонстрировал высокую скорость обработки благодаря вычислениям в оперативной памяти и оптимизированному выполнению как RDD-операций, так и SQL-запросов.