МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт компьютерных наук
Кафедра автоматизированных систем управления

Лабораторная работа №1

по дисциплине «Теория обработки больших массивов данных»

Студент М-РИТ-25 _____ Станиславчук С. М.

(подпись, дата)

Руководитель

Доцент, к.т.н. _____ Тюрин А. С.

(подпись, дата)

Липецк 2025

Содержание

1. Задание кафедры

Сконфигурировать Docker (Compose) со Spark.
Цель: протестировать кооперативную работу мастера и воркеров, путем запуска Python-скрипта (тестов) внутри контейнера.

2. Ход работы

** В ходе выполнения работы использовалось ядро GNU/Linux v6.15.4, Docker v28.3.0, Spark v3.5.4, OpenJDK v17.0.14, Python v3.12.9 **

Краткий разбор docker-compose.yml

Конфигурация мастера:

```
  spark-master:
    image: bitnami/spark:3.5.4 # источник загруки изображения спарка
    container_name: spark-master
    environment:
      - SPARK_MODE=master # версия нода
# Отключаем ненужный (в ходе данной работы) функционал
      - SPARK_RPC_AUTHENTICATION_ENABLED=no
      - SPARK_RPC_ENCRYPTION_ENABLED=no
      - SPARK_LOCAL_STORAGE_ENCRYPTION_ENABLED=no
      - SPARK_SSL_ENABLED=no
      - SPARK_MASTER_HOST=spark-master # явно указываем имя мастера-хоста
    ports: # используемые порты
      - "8080:8080"
      - "7077:7077"
    networks:
      - spark-net # network для работы с сетью (опционально)
```

Конфигурация воркера:

```
  spark-worker:
    image: bitnami/spark:3.5.4
    container_name: spark-worker
    environment:
      - SPARK_MODE=worker
      - SPARK_MASTER_URL=spark://spark-master:7077 # используемый URL
      - SPARK_WORKER_CORES=2 # выделенное кол-во ядер системы
      - SPARK_WORKER_MEMORY=2g # выделенное кол-во памяти
      - SPARK_WORKER_WEBUI_PORT=8081 # порт для отображения WEBUI воркера
    ports:
      - "8081:8081"
    depends_on:
      - spark-master # сначала загрузится spark-master, а затем только worker
    networks:
```

```
    - spark-net
```

Следующий шаг: запустить оба контейнера. Запуск представлен на рис. 1

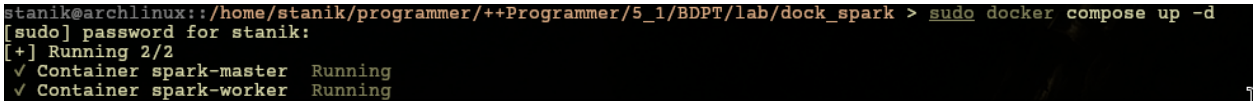

Рисунок 1 — Выполнение команды up

Python-скрипт, который будет передан мастеру и в следствие запущен на воркере, представлен в приложении 2. Это тест-скрипт, который имеет несколько простых функций (каждая из них будет являться JOB’ом в логах после выполнения).

Для того, чтобы мастер-контейнер мог работать со скриптом, его нужно вручную отправить в изолированный контейнер. Команда копирования скрипта в контейнер «spark-master» представлена на рисунке 2.

Рисунок 2 - docker cp pscript.py spark-master:/opt/bitnami/spark/script.py

Остается только запустить скрипт и наблюдать результат в логах.

Команда запуска скрипта: `docker exec -it spark-master /opt/bitnami/spark/bin/spark-submit --master spark://spark-master:7077 /opt/bitnami/spark/script.py`

После аргументов -it указываем контейнер, в котором будем работать, путь и затем само тело команды — это исполняемый spark-submit, с аргументом — master и путем до самого исполняемого агента (на порте 7077, указанного в конфиге), ну и указываем сам путь до скопированного скрипта.

В первую очередь протестировали коннект между local и Docker Spark. После успешного коннекта, прошли сами rdd-тесты. По итогу все тесты были пройдены и спарк автоматически (по инструктии .py скрипта) был

остановлен.

Дисплей панели администратора Spark-Worker’а представлена на рисунках 3 и 4.

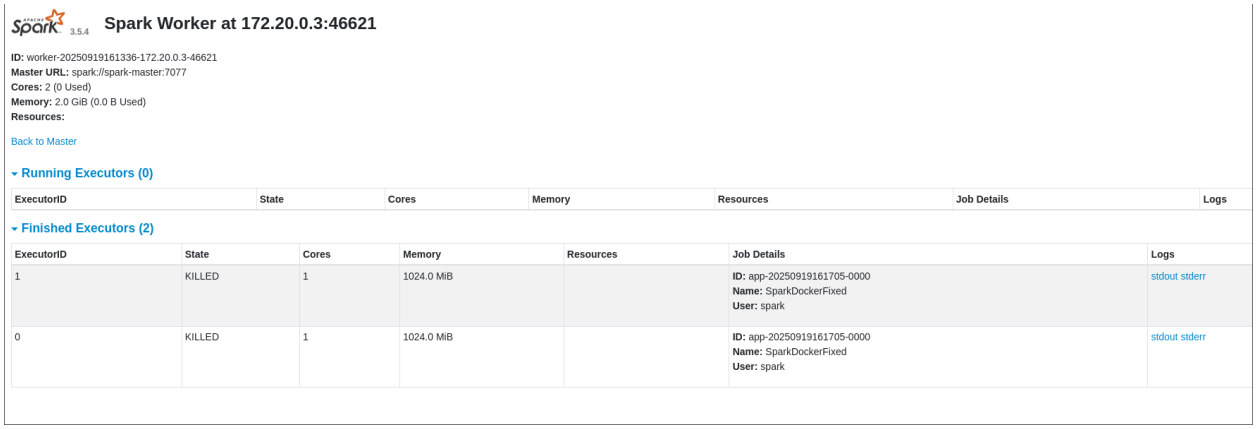

Рисунок 3 — Панель администратора Spark (Worker-node). С законченными работами.

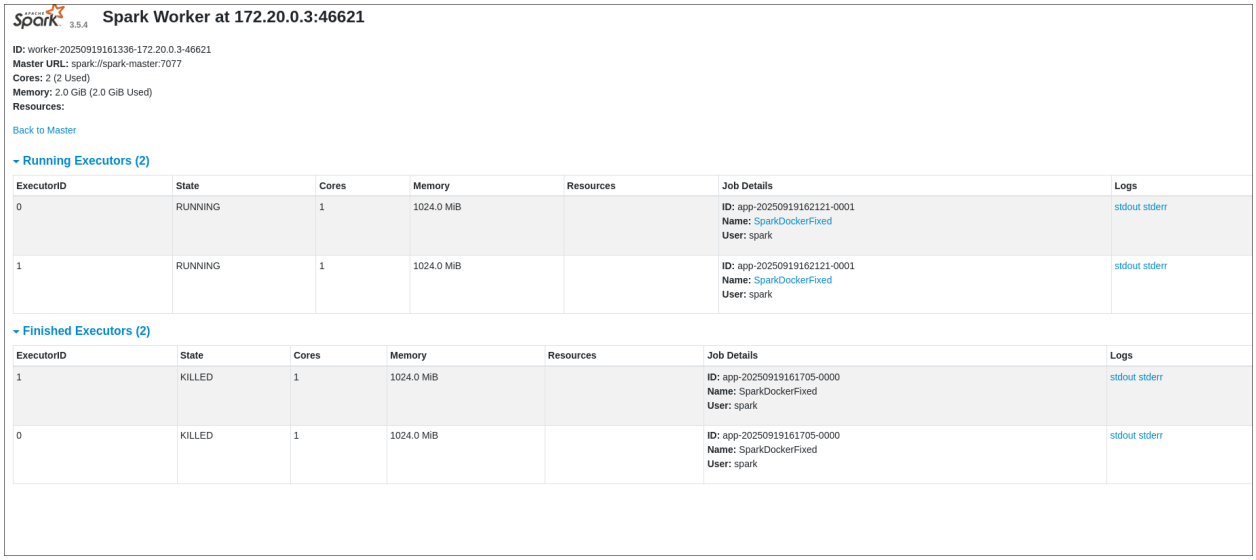

Рисунок 4 — Панель администратора Spark (Worker-node). С выполняемыми работами.

Здесь ExecutorID — это id процесса, который выполняет поставленную задачу.

В Job Details:

- ID — это id работы.

- Name — имя работы (задавали в скрипте)
- User — spark, так как сам скрипт запускался внутри оболочки самого spark-master, пользователь имел именно это имя.

Сначала были выполнены работы с нодами, которые я не успел зафиксировать, поэтому на рисунке 4 уже есть 2 выполненные работы. Полный результат (лог выполнения скрипта) приведен в приложении 3.

3. Вывод

В ходе выполненной работы были запущены докер-контейнеры со Spark-worker и Spark-master, был выполнен python-скрипт и проанализированы результаты его выполнения.

Приложение 1. docker-compose.yml

```
services:
  spark-master:
    image: bitnami/spark:3.5.4 # отсюда загрузится изображение спарка
    container_name: spark-master
    environment:
      - SPARK_MODE=master # версия нода
# Отключаем ненужный (в ходе данной работы) функционал
      - SPARK_RPC_AUTHENTICATION_ENABLED=no
      - SPARK_RPC_ENCRYPTION_ENABLED=no
      - SPARK_LOCAL_STORAGE_ENCRYPTION_ENABLED=no
      - SPARK_SSL_ENABLED=no
      - SPARK_MASTER_HOST=spark-master # явно указываем имя мастера-хоста
    ports: # используемые порты
      - "8080:8080"
      - "7077:7077"
    networks:
      - spark-net # network для работы с сетью (опционально)

  spark-worker:
    image: bitnami/spark:3.5.4
    container_name: spark-worker
    environment:
      - SPARK_MODE=worker
      - SPARK_MASTER_URL=spark://spark-master:7077
      - SPARK_WORKER_CORES=2
      - SPARK_WORKER_MEMORY=2g
      - SPARK_WORKER_WEBUI_PORT=8081
    ports:
      - "8081:8081"
    depends_on:
      - spark-master
    networks:
      - spark-net

networks:
  spark-net:
    driver: bridge
    ipam:
      config:
        - subnet: 172.20.0.0/16
```

Приложение 2. Python-скрипт

```
import time
from pyspark.sql import SparkSession
from pyspark import SparkConf
import socket

def get_host_ip():
    """Get host IP that containers can access"""
    try:
        # Connect to Spark master to determine reachable IP
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        s.connect(("spark-master", 7077))
        ip = s.getsockname()[0]
        s.close()
        return ip
    except:
        # Fallback to host.docker.internal for Mac
        return "host.docker.internal"

def create_spark_session():
    """Create Spark session that works with Docker networking"""

    driver_host = get_host_ip()
    print(f"Using driver host: {driver_host}")

    conf = SparkConf()
    conf.setAppName("SparkDockerFixed")
    conf.setMaster("spark://spark-master:7077")
    conf.set("spark.driver.host", driver_host)
    conf.set("spark.driver.port", "4050")
    conf.set("spark.driver.bindAddress", "0.0.0.0")
    conf.set("spark.executor.memory", "1g")
    conf.set("spark.executor.cores", "1")
    conf.set("spark.network.timeout", "300s")
    conf.set("spark.sql.adaptive.enabled", "true")

    spark = SparkSession.builder.config(conf=conf).getOrCreate()
    return spark

# Использование
spark = create_spark_session()

def simple_operation_test(spark):
    """Test simple operations without complex transformations"""

    print("Testing basic RDD operations...")

    # Test 1: Simple count
    try:
        rdd = spark.sparkContext.parallelize([1, 2, 3, 4, 5], 2)  # 2 partitions
        count = rdd.count() # JOB #0
        print(f"  Count test passed: {count}")
    except Exception as e:
        print(f"  Count test failed: {e}")
        return False
```

```
    # Test 2: Simple collect (avoid complex operations initially)
    try:
        rdd = spark.sparkContext.parallelize([1, 2, 3], 1)
        data = rdd.collect() # JOB #1
        print(f"  Collect test passed: {data}")
    except Exception as e:
        print(f"  Collect test failed: {e}")
        return False

    # Test 3: Simple map
    try:
        rdd = spark.sparkContext.parallelize([1, 2, 3], 1)
        mapped = rdd.map(lambda x: x * 2)
        result = mapped.collect() # JOB #1
        print(f"  Map test passed: {result}")
    except Exception as e:
        print(f"  Map test failed: {e}")
        return False

    # Only after basic tests work, try sum
    try:
        rdd = spark.sparkContext.parallelize([1, 2, 3, 4, 5], 2)
        result = rdd.sum() # JOB #3
        print(f"  Sum test passed: {result}")
    except Exception as e:
        print(f"  Sum test failed: {e}")
        print("Trying alternative sum implementation...")

        # Alternative approach for sum
        try:
            rdd = spark.sparkContext.parallelize([1, 2, 3, 4, 5], 2)
            result = rdd.reduce(lambda a, b: a + b)
            print(f"  Reduce sum passed: {result}")
        except Exception as e:
            print(f"  Reduce sum also failed: {e}")
            return False

    return True

def main():
    print("  Starting local to Docker Spark connection test...")
    print("Waiting 10 seconds for cluster to be ready...")
    time.sleep(10)

    try:
        spark = create_spark_session()
        print("  Spark session created successfully!")

        # Test basic functionality
        if simple_operation_test(spark):
            print("\n  All tests passed! Connection is working.")
        else:
            print("\n  Some tests failed, but connection is established.")

        # Show cluster info
        print(f"\n  Cluster information:")
```

```
        print(f"Master URL: {spark.sparkContext.master}")
        print(f"Application ID: {spark.sparkContext.applicationId}")
        print(f"Spark UI: http://localhost:4040")

    except Exception as e:
        print(f"  Failed to create Spark session: {e}")
        import traceback
        traceback.print_exc()

    finally:
        try:
            # spark.stop()
            print("Spark session stopped.")
        except:
            pass

if __name__ == "__main__":
    main()
```

Приложение 3. Лог выполнения

```
Using driver host: 172.20.0.2
25/09/05 18:14:36 INFO SparkContext: Running Spark version 3.5.4
25/09/05 18:14:36 INFO SparkContext: OS info Linux, 6.15.4-arch2-1, amd64
25/09/05 18:14:36 INFO SparkContext: Java version 17.0.14
25/09/05 18:14:36 INFO ResourceUtils:
==============================================================
25/09/05 18:14:36 INFO ResourceUtils: No custom resources configured for spark.driver.
25/09/05 18:14:36 INFO ResourceUtils:
==============================================================
25/09/05 18:14:36 INFO SparkContext: Submitted application: SparkDockerFixed
25/09/05 18:14:36 INFO ResourceProfile: Default ResourceProfile created, executor resources:
Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024,
script: , vendor: , offHe
ap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus,
amount: 1.0)
25/09/05 18:14:36 INFO ResourceProfile: Limiting resource is cpus at 1 tasks per executor
25/09/05 18:14:36 INFO ResourceProfileManager: Added ResourceProfile id: 0
25/09/05 18:14:36 INFO SecurityManager: Changing view acls to: spark
25/09/05 18:14:36 INFO SecurityManager: Changing modify acls to: spark
25/09/05 18:14:36 INFO SecurityManager: Changing view acls groups to:
25/09/05 18:14:36 INFO SecurityManager: Changing modify acls groups to:
25/09/05 18:14:36 INFO SecurityManager: SecurityManager: authentication disabled; ui acls
disabled; users with view permissions: spark; groups with view permissions: EMPTY; users with
modify permissions: spark; gr
oups with modify permissions: EMPTY
25/09/05 18:14:36 WARN NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
25/09/05 18:14:36 INFO Utils: Successfully started service 'sparkDriver' on port 4050.
25/09/05 18:14:36 INFO SparkEnv: Registering MapOutputTracker
25/09/05 18:14:36 INFO SparkEnv: Registering BlockManagerMaster
25/09/05 18:14:36 INFO BlockManagerMasterEndpoint: Using
org.apache.spark.storage.DefaultTopologyMapper for getting topology information
25/09/05 18:14:36 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
25/09/05 18:14:36 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
25/09/05 18:14:36 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-46cf7059-c0a2-
442d-91e3-42e17690e2e5
25/09/05 18:14:36 INFO MemoryStore: MemoryStore started with capacity 434.4 MiB
25/09/05 18:14:36 INFO SparkEnv: Registering OutputCommitCoordinator
25/09/05 18:14:36 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for SparkUI
25/09/05 18:14:36 INFO Utils: Successfully started service 'SparkUI' on port 4040.
25/09/05 18:14:37 INFO StandaloneAppClient$ClientEndpoint: Connecting to master spark://spark-
master:7077...
25/09/05 18:14:37 INFO TransportClientFactory: Successfully created connection to
spark-master/172.20.0.2:7077 after 24 ms (0 ms spent in bootstraps)
25/09/05 18:14:37 INFO StandaloneSchedulerBackend: Connected to Spark cluster with app ID
app-20250905181437-0002
25/09/05 18:14:37 INFO StandaloneAppClient$ClientEndpoint: Executor added: app-20250905181437-
0002/0 on worker-20250905164554-172.20.0.3-41797 (172.20.0.3:41797) with 1 core(s)
25/09/05 18:14:37 INFO StandaloneSchedulerBackend: Granted executor ID app-20250905181437-0002/0
on hostPort 172.20.0.3:41797 with 1 core(s), 1024.0 MiB RAM
25/09/05 18:14:37 INFO StandaloneAppClient$ClientEndpoint: Executor added: app-20250905181437-
0002/1 on worker-20250905164554-172.20.0.3-41797 (172.20.0.3:41797) with 1 core(s)
25/09/05 18:14:37 INFO StandaloneSchedulerBackend: Granted executor ID app-20250905181437-0002/1
on hostPort 172.20.0.3:41797 with 1 core(s), 1024.0 MiB RAM
```

```
25/09/05 18:14:37 INFO Utils: Successfully started service
'org.apache.spark.network.netty.NettyBlockTransferService' on port 41275.
25/09/05 18:14:37 INFO NettyBlockTransferService: Server created on 172.20.0.2 0.0.0.0:41275
25/09/05 18:14:37 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy
for block replication policy
25/09/05 18:14:37 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver,
172.20.0.2, 41275, None)
25/09/05 18:14:37 INFO BlockManagerMasterEndpoint: Registering block manager 172.20.0.2:41275
with 434.4 MiB RAM, BlockManagerId(driver, 172.20.0.2, 41275, None)
25/09/05 18:14:37 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver,
172.20.0.2, 41275, None)
25/09/05 18:14:37 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 172.20.0.2,
41275, None)
25/09/05 18:14:37 INFO StandaloneAppClient$ClientEndpoint: Executor updated: app-20250905181437-
0002/0 is now RUNNING
25/09/05 18:14:37 INFO StandaloneAppClient$ClientEndpoint: Executor updated: app-20250905181437-
0002/1 is now RUNNING
25/09/05 18:14:37 INFO StandaloneSchedulerBackend: SchedulerBackend is ready for scheduling
beginning after reached minRegisteredResourcesRatio: 0.0
  Starting local to Docker Spark connection test...
Waiting 10 seconds for cluster to be ready...
25/09/05 18:14:39 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Registered executor
NettyRpcEndpointRef(spark-client://Executor) (172.20.0.3:38334) with ID 1,  ResourceProfileId 0
25/09/05 18:14:39 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Registered executor
NettyRpcEndpointRef(spark-client://Executor) (172.20.0.3:38348) with ID 0,  ResourceProfileId 0
25/09/05 18:14:39 INFO BlockManagerMasterEndpoint: Registering block manager 172.20.0.3:41849
with 434.4 MiB RAM, BlockManagerId(1, 172.20.0.3, 41849, None)
25/09/05 18:14:39 INFO BlockManagerMasterEndpoint: Registering block manager 172.20.0.3:45401
with 434.4 MiB RAM, BlockManagerId(0, 172.20.0.3, 45401, None)
Using driver host: 172.20.0.2
25/09/05 18:14:47 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of
spark.sql.warehouse.dir.
25/09/05 18:14:47 INFO SharedState: Warehouse path is 'file:/opt/bitnami/spark/spark-warehouse'.
25/09/05 18:14:48 WARN SparkSession: Using an existing Spark session; only runtime SQL
configurations will take effect.
  Spark session created successfully!
21:14:50 [56/1250]
Testing basic RDD operations...
25/09/05 18:14:48 INFO SparkContext: Starting job: count at /opt/bitnami/spark/script.py:50
25/09/05 18:14:48 INFO DAGScheduler: Got job 0 (count at /opt/bitnami/spark/script.py:50) with 2
output partitions
25/09/05 18:14:48 INFO DAGScheduler: Final stage: ResultStage 0 (count at
/opt/bitnami/spark/script.py:50)
25/09/05 18:14:48 INFO DAGScheduler: Parents of final stage: List()
25/09/05 18:14:48 INFO DAGScheduler: Missing parents: List()
25/09/05 18:14:48 INFO DAGScheduler: Submitting ResultStage 0 (PythonRDD[1] at count at
/opt/bitnami/spark/script.py:50), which has no missing parents
25/09/05 18:14:48 INFO MemoryStore: Block broadcast_0 stored as values in memory (estimated size
7.5 KiB, free 434.4 MiB)
25/09/05 18:14:48 INFO MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated
size 4.8 KiB, free 434.4 MiB)
25/09/05 18:14:48 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on 172.20.0.2:41275
(size: 4.8 KiB, free: 434.4 MiB)
25/09/05 18:14:48 INFO SparkContext: Created broadcast 0 from broadcast at
DAGScheduler.scala:1585
25/09/05 18:14:48 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 0 (PythonRDD[1]
at count at /opt/bitnami/spark/script.py:50) (first 15 tasks are for partitions Vector(0, 1))
```

```
25/09/05 18:14:48 INFO TaskSchedulerImpl: Adding task set 0.0 with 2 tasks resource profile 0
25/09/05 18:14:48 INFO TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0) (172.20.0.3,
executor 0, partition 0, PROCESS_LOCAL, 9013 bytes)
25/09/05 18:14:48 INFO TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1) (172.20.0.3,
executor 1, partition 1, PROCESS_LOCAL, 9040 bytes)
25/09/05 18:14:48 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on 172.20.0.3:45401
(size: 4.8 KiB, free: 434.4 MiB)
25/09/05 18:14:48 INFO BlockManagerInfo: Added broadcast_0_piece0 in memory on 172.20.0.3:41849
(size: 4.8 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 1585 ms on
172.20.0.3 (executor 1) (1/2)
25/09/05 18:14:50 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1
port: 45591
25/09/05 18:14:50 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 1620 ms on
172.20.0.3 (executor 0) (2/2)
25/09/05 18:14:50 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed,
from pool
25/09/05 18:14:50 INFO DAGScheduler: ResultStage 0 (count at /opt/bitnami/spark/script.py:50)
finished in 1.726 s
25/09/05 18:14:50 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or
zombie tasks for this job
25/09/05 18:14:50 INFO TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
25/09/05 18:14:50 INFO DAGScheduler: Job 0 finished: count at /opt/bitnami/spark/script.py:50,
took 1.762776 s
  Count test passed: 5
25/09/05 18:14:50 INFO SparkContext: Starting job: collect at /opt/bitnami/spark/script.py:59
25/09/05 18:14:50 INFO DAGScheduler: Got job 1 (collect at /opt/bitnami/spark/script.py:59) with
1 output partitions
25/09/05 18:14:50 INFO DAGScheduler: Final stage: ResultStage 1 (collect at
/opt/bitnami/spark/script.py:59)
25/09/05 18:14:50 INFO DAGScheduler: Parents of final stage: List()
25/09/05 18:14:50 INFO DAGScheduler: Missing parents: List()
25/09/05 18:14:50 INFO DAGScheduler: Submitting ResultStage 1 (ParallelCollectionRDD[2] at
readRDDFromFile at PythonRDD.scala:289), which has no missing parents
25/09/05 18:14:50 INFO MemoryStore: Block broadcast_1 stored as values in memory (estimated size
2.9 KiB, free 434.4 MiB)
25/09/05 18:14:50 INFO MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated
size 1757.0 B, free 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Added broadcast_1_piece0 in memory on 172.20.0.2:41275
(size: 1757.0 B, free: 434.4 MiB)
25/09/05 18:14:50 INFO SparkContext: Created broadcast 1 from broadcast at
DAGScheduler.scala:1585
25/09/05 18:14:50 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 1
(ParallelCollectionRDD[2] at readRDDFromFile at PythonRDD.scala:289) (first 15 tasks are for
partitions Vector(0))
25/09/05 18:14:50 INFO TaskSchedulerImpl: Adding task set 1.0 with 1 tasks resource profile 0
25/09/05 18:14:50 INFO TaskSetManager: Starting task 0.0 in stage 1.0 (TID 2) (172.20.0.3,
executor 1, partition 0, PROCESS_LOCAL, 9015 bytes)
25/09/05 18:14:50 INFO BlockManagerInfo: Added broadcast_1_piece0 in memory on 172.20.0.3:41849
(size: 1757.0 B, free: 434.4 MiB)
25/09/05 18:14:50 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 2) in 28 ms on
172.20.0.3 (executor 1) (1/1)
25/09/05 18:14:50 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed,
from pool
25/09/05 18:14:50 INFO DAGScheduler: ResultStage 1 (collect at /opt/bitnami/spark/script.py:59)
finished in 0.038 s
25/09/05 18:14:50 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or
```

```
zombie tasks for this job
25/09/05 18:14:50 INFO TaskSchedulerImpl: Killing all running tasks in stage 1: Stage finished
25/09/05 18:14:50 INFO DAGScheduler: Job 1 finished: collect at /opt/bitnami/spark/script.py:59,
took 0.044012 s
  Collect test passed: [1, 2, 3]
25/09/05 18:14:50 INFO SparkContext: Starting job: collect at /opt/bitnami/spark/script.py:69
25/09/05 18:14:50 INFO DAGScheduler: Got job 2 (collect at /opt/bitnami/spark/script.py:69) with
1 output partitions
25/09/05 18:14:50 INFO DAGScheduler: Final stage: ResultStage 2 (collect at
/opt/bitnami/spark/script.py:69)
25/09/05 18:14:50 INFO DAGScheduler: Parents of final stage: List()
25/09/05 18:14:50 INFO DAGScheduler: Missing parents: List()
25/09/05 18:14:50 INFO DAGScheduler: Submitting ResultStage 2 (PythonRDD[4] at collect at
/opt/bitnami/spark/script.py:69), which has no missing parents
25/09/05 18:14:50 INFO MemoryStore: Block broadcast_2 stored as values in memory (estimated size
5.7 KiB, free 434.4 MiB)
25/09/05 18:14:50 INFO MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated
size 3.7 KiB, free 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Added broadcast_2_piece0 in memory on 172.20.0.2:41275
(size: 3.7 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO SparkContext: Created broadcast 2 from broadcast at
DAGScheduler.scala:1585
25/09/05 18:14:50 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 2 (PythonRDD[4]
at collect at /opt/bitnami/spark/script.py:69) (first 15 tasks are for partitions Vector(0))
25/09/05 18:14:50 INFO TaskSchedulerImpl: Adding task set 2.0 with 1 tasks resource profile 0
25/09/05 18:14:50 INFO TaskSetManager: Starting task 0.0 in stage 2.0 (TID 3) (172.20.0.3,
executor 0, partition 0, PROCESS_LOCAL, 9015 bytes)
25/09/05 18:14:50 INFO BlockManagerInfo: Added broadcast_2_piece0 in memory on 172.20.0.3:45401
(size: 3.7 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO TaskSetManager: Finished task 0.0 in stage 2.0 (TID 3) in 111 ms on
172.20.0.3 (executor 0) (1/1)
25/09/05 18:14:50 INFO TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed,
from pool
25/09/05 18:14:50 INFO DAGScheduler: ResultStage 2 (collect at /opt/bitnami/spark/script.py:69)
finished in 0.121 s
25/09/05 18:14:50 INFO DAGScheduler: Job 2 is finished. Cancelling potential speculative or
zombie tasks for this job
25/09/05 18:14:50 INFO TaskSchedulerImpl: Killing all running tasks in stage 2: Stage finished
25/09/05 18:14:50 INFO DAGScheduler: Job 2 finished: collect at /opt/bitnami/spark/script.py:69,
took 0.125173 s

  Map test passed: [2, 4, 6]
25/09/05 18:14:50 INFO SparkContext: Starting job: sum at /opt/bitnami/spark/script.py:78
25/09/05 18:14:50 INFO DAGScheduler: Got job 3 (sum at /opt/bitnami/spark/script.py:78) with 2
output partitions
25/09/05 18:14:50 INFO DAGScheduler: Final stage: ResultStage 3 (sum at
/opt/bitnami/spark/script.py:78)
25/09/05 18:14:50 INFO DAGScheduler: Parents of final stage: List()
25/09/05 18:14:50 INFO DAGScheduler: Missing parents: List()
25/09/05 18:14:50 INFO DAGScheduler: Submitting ResultStage 3 (PythonRDD[6] at sum at
/opt/bitnami/spark/script.py:78), which has no missing parents
25/09/05 18:14:50 INFO MemoryStore: Block broadcast_3 stored as values in memory (estimated size
6.9 KiB, free 434.4 MiB)
25/09/05 18:14:50 INFO MemoryStore: Block broadcast_3_piece0 stored as bytes in memory (estimated
size 4.5 KiB, free 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Added broadcast_3_piece0 in memory on 172.20.0.2:41275
(size: 4.5 KiB, free: 434.4 MiB)
```

```
25/09/05 18:14:50 INFO SparkContext: Created broadcast 3 from broadcast at
DAGScheduler.scala:1585
25/09/05 18:14:50 INFO DAGScheduler: Submitting 2 missing tasks from ResultStage 3 (PythonRDD[6]
at sum at /opt/bitnami/spark/script.py:78) (first 15 tasks are for partitions Vector(0, 1))
25/09/05 18:14:50 INFO TaskSchedulerImpl: Adding task set 3.0 with 2 tasks resource profile 0
25/09/05 18:14:50 INFO TaskSetManager: Starting task 0.0 in stage 3.0 (TID 4) (172.20.0.3,
executor 1, partition 0, PROCESS_LOCAL, 9013 bytes)
25/09/05 18:14:50 INFO TaskSetManager: Starting task 1.0 in stage 3.0 (TID 5) (172.20.0.3,
executor 0, partition 1, PROCESS_LOCAL, 9040 bytes)
25/09/05 18:14:50 INFO BlockManagerInfo: Removed broadcast_0_piece0 on 172.20.0.2:41275 in memory
(size: 4.8 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Added broadcast_3_piece0 in memory on 172.20.0.3:41849
(size: 4.5 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Added broadcast_3_piece0 in memory on 172.20.0.3:45401
(size: 4.5 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Removed broadcast_0_piece0 on 172.20.0.3:41849 in memory
(size: 4.8 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Removed broadcast_0_piece0 on 172.20.0.3:45401 in memory
(size: 4.8 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Removed broadcast_2_piece0 on 172.20.0.2:41275 in memory
(size: 3.7 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Removed broadcast_2_piece0 on 172.20.0.3:45401 in memory
(size: 3.7 KiB, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Removed broadcast_1_piece0 on 172.20.0.2:41275 in memory
(size: 1757.0 B, free: 434.4 MiB)
25/09/05 18:14:50 INFO BlockManagerInfo: Removed broadcast_1_piece0 on 172.20.0.3:41849 in memory
(size: 1757.0 B, free: 434.4 MiB)
25/09/05 18:14:50 INFO TaskSetManager: Finished task 1.0 in stage 3.0 (TID 5) in 120 ms on
172.20.0.3 (executor 0) (1/2)
25/09/05 18:14:50 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 4) in 125 ms on
172.20.0.3 (executor 1) (2/2)
25/09/05 18:14:50 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed,
from pool
25/09/05 18:14:50 INFO DAGScheduler: ResultStage 3 (sum at /opt/bitnami/spark/script.py:78)
finished in 0.139 s
25/09/05 18:14:50 INFO DAGScheduler: Job 3 is finished. Cancelling potential speculative or
zombie tasks for this job
25/09/05 18:14:50 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
25/09/05 18:14:50 INFO DAGScheduler: Job 3 finished: sum at /opt/bitnami/spark/script.py:78, took
0.144019 s
  Sum test passed: 15

  All tests passed! Connection is working.

  Cluster information:
Master URL: spark://spark-master:7077
Application ID: app-20250905181437-0002
Spark UI: http://localhost:4040
Spark session stopped.
25/09/05 18:14:50 INFO SparkContext: Invoking stop() from shutdown hook
25/09/05 18:14:50 INFO SparkContext: SparkContext is stopping with exitCode 0.
25/09/05 18:14:50 INFO SparkUI: Stopped Spark web UI at http://172.20.0.2:4040
25/09/05 18:14:50 INFO StandaloneSchedulerBackend: Shutting down all executors
25/09/05 18:14:50 INFO StandaloneSchedulerBackend$StandaloneDriverEndpoint: Asking each executor
to shut down
25/09/05 18:14:50 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/09/05 18:14:50 INFO MemoryStore: MemoryStore cleared
```

```
25/09/05 18:14:50 INFO BlockManager: BlockManager stopped
25/09/05 18:14:50 INFO BlockManagerMaster: BlockManagerMaster stopped
25/09/05 18:14:50 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:
OutputCommitCoordinator stopped!
25/09/05 18:14:50 INFO SparkContext: Successfully stopped SparkContext
25/09/05 18:14:50 INFO ShutdownHookManager: Shutdown hook called
25/09/05 18:14:50 INFO ShutdownHookManager: Deleting directory /tmp/spark-6ad982a2-ebd1-4910-
91d4-7cdaa9921ba4
25/09/05 18:14:50 INFO ShutdownHookManager: Deleting directory /tmp/spark-19dad691-9bc5-4d5e-
a90f-b0766a0a1280/pyspark-82e2a426-f386-4fc6-ab71-4725885352d4
25/09/05 18:14:50 INFO ShutdownHookManager: Deleting directory /tmp/spark-19dad691-9bc5-4d5e-
a90f-b0766a0a1280
```