

СОЗДАНИЕ КЛАСТЕРА ИЗ ВИРТУАЛЬНЫХ МАШИН С ЦЕЛЬЮ СОЗДАНИЯ МОДЕЛИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ MAPREDUCE

Стафеев И.А., Голованов Д.И., Шарыпов Е.А (Университет ИТМО)
Научный руководитель - кандидат технических наук, доцент Добриборщ
Д.Э. (Университет ИТМО)

ВВЕДЕНИЕ.

В рамках семестрового курса по информатике нами были получены теоретические и небольшие практические знания в областях Unix-систем, языка программирования Bash, работы с терминалом, компьютерных сетей, виртуализации и контейнеризации. Смысл разработанной лабораторной работы - обобщить полученные знания посредством их комплексного применения для решения одной практической задачи. В качестве такой задачи мы выбрали развертывание кластера из виртуальных машин с установленной на них распределенной файловой системой, чтобы созданный кластер мог служить средством обработки больших данных в соответствии с парадигмой распределенных вычислений MapReduce.

Для достижения поставленной цели (создания кластера) были поставлены следующие задачи:

1. придумать практическую задачу, решение которой может быть осуществлено с помощью кластера
2. создать и настроить виртуальные машины, чтобы на них можно было установить распределенную файловую систему
3. установить распределенную файловую систему на виртуальные машины, тем самым создав кластер
4. написать скрипт(-ы), которые могут решить придуманную задачу и которые соответствуют парадигме MapReduce
5. провести тестовый и несколько рабочих прогонов входных данных задачи, чтобы доказать корректность работы кластера и написанных скриптов
6. на каждом этапе проводить выявление и устранение возможных ошибок и недочетов

В этом отчете представлены теоретические выкладки относительно упомянутой парадигмы, поэтапное описание лабораторной работы с описанием действий для ее выполнения, описание знаний и навыков, используемых или получаемых в процессе выполнения работы и результат выполнения лабораторной работы членами команды.

ОСНОВНАЯ ЧАСТЬ.

Теоретические выкладки. В настоящее время в связи с растущим объемом информации, производимой участниками какой-либо сетевой архитектуры к обработке

этой информации предъявляются требования параллельности, горизонтальной масштабируемости, отказоустойчивости и локальности данных [1]. Для удовлетворения этих целей все чаще применяется создание кластеров - групп вычислительных устройств, объединенных в одну сеть использующихся как единый ресурс и обеспечивающих физически объединенное хранение данных из разных источников. В таких кластерах используются распределенные файловые системы, то есть такие, которые могут дробить единые данные на блоки и хранить их на разных серверах в сети. Именно этой вещи и посвящена лабораторная работа. То есть она призвана не только закрепить знания, полученные в рамках курса, но и дать новые, которые посвящены вещам, которые активно используются в настоящее время и которые вследствие могут потенциально быть полезными в дальнейшей профессиональной деятельности.

Конечно, только хранить данные на серверах почти бессмысленно, они должны подвергаться обработке. Здесь будет рассмотрена одна из наиболее известных моделей распределенной обработки данных - MapReduce [1, 2].

Эта парадигма вычислений предполагает 3 стадии обработки данных (см. картинку 1):

1. Map. На этой стадии все данные в виде блоков проходят через функцию-отображение, которая преобразует каждый элемент в пару "ключ-значение". Важно отметить, что функция map обычно применяется на той же машине, на которой хранятся данные, что позволяет не тратить время на передачу данных между устройствами.
2. Shuffle. Полученные на предыдущей стадии пары значений распределяются в условные корзины по ключу для дальнейшей обработки
3. Reduce. На этой стадии происходят данные с одним ключом проходят через функцию-свертку, где над ними происходят какие-то действия. Результат работы третьей стадии — это результат работы всей MapReduce-задачи.

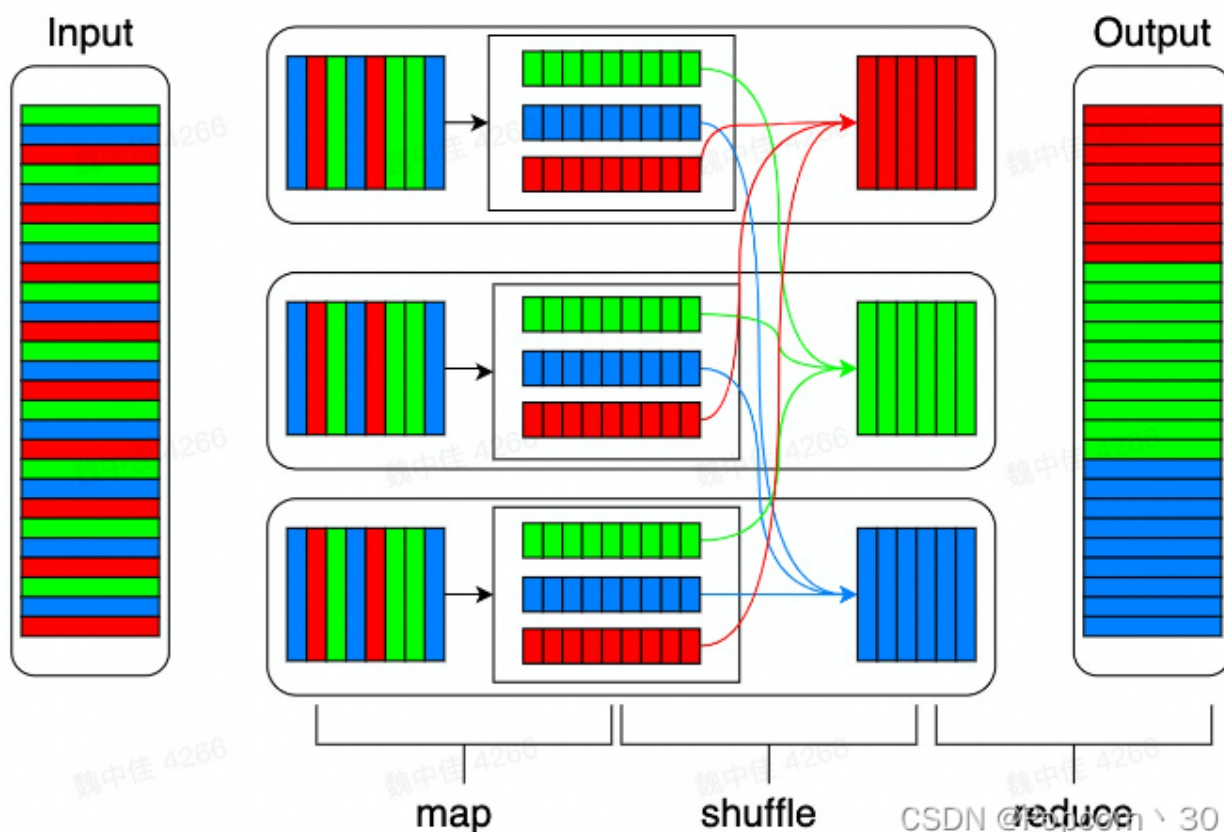


Рисунок 1 — Устройство парадигмы MapReduce

Для создания кластера и вычислений в соответствии с описанной моделью мы решили использовать технологии Apache Software Foundation, такие как Hadoop, YARN и Airflow. Выбор в пользу разработок этой организации был сделан, во-первых, потому что это open-source проекта, и во-вторых, так как мы имеем небольшой опыт работы с ними.

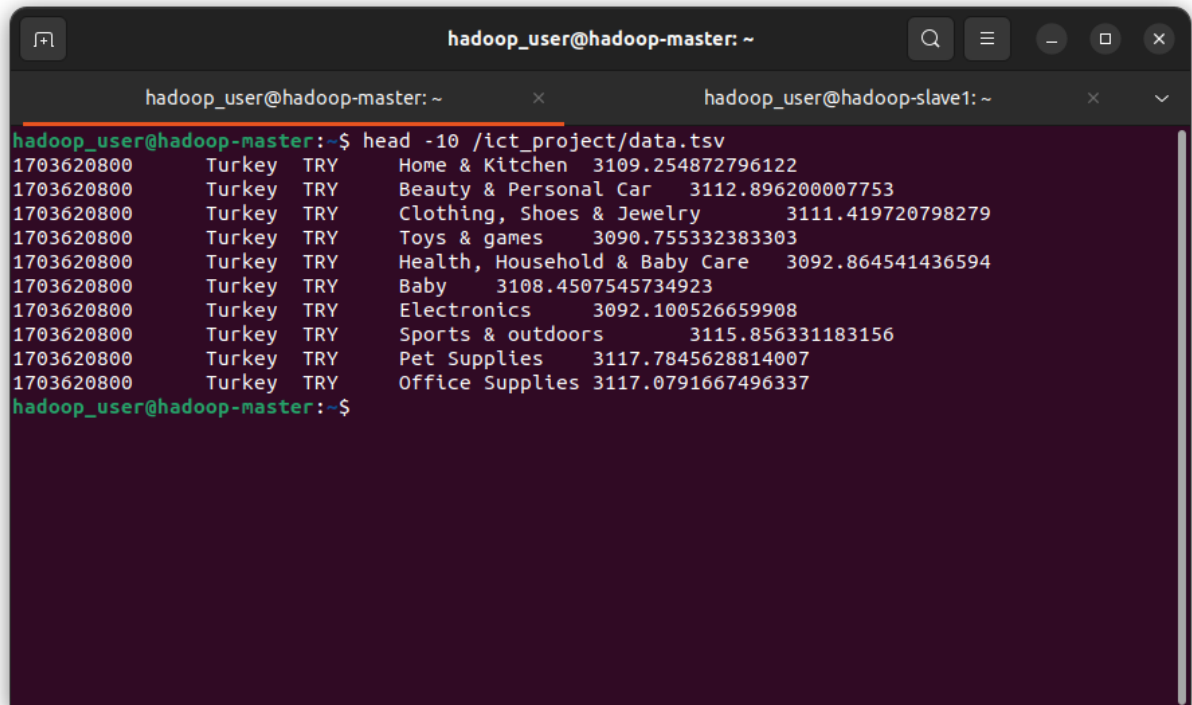
Hadoop [3] — это инструмент для распределенного хранения данных (в HDFS - Hadoop Distributed File System) и запуска MapReduce-задач над ними посредством использования YARN

YARN (Yet Another Resource Negotiator) [4] — это система планирования заданий и управления кластером, набор системных программ, обеспечивающих совместное использование, масштабирование и надежность работы распределенных приложений

Airflow [5] — это инструмент для оркестрации данных. В Airflow существует основное понятие - DAG (Directed Acyclic Graph). Это концептуальное представление серии действий в виде графа, в котором данные могут следовать только в одном направлении.

Мы предполагаем следующий процесс использования этих технологий. После первичной настройки виртуальных машин необходимо развернуть на них HDFS и YARN, с помощью которых провести процесс преобразования данных, а затем сделать этот процесс запланированным, чтобы необходимые операции с помощью Airflow проводились с некоторой периодичностью.

Постановка задачи. Теперь обозначим, какую задачу мы призваны решить с помощью кластера. Пусть имеется датасет, содержащих данные о покупках - час совершения, регион, категорию товара, валюту, в которой совершена покупка и стоимость в этой валюте (см. рисунок 2).



```
hadoop_user@hadoop-master: ~  
hadoop_user@hadoop-master: ~  
hadoop_user@hadoop-master: ~  
hadoop_user@hadoop-master: ~$ head -10 /ict_project/data.tsv  
1703620800 Turkey TRY Home & Kitchen 3109.254872796122  
1703620800 Turkey TRY Beauty & Personal Car 3112.896200007753  
1703620800 Turkey TRY Clothing, Shoes & Jewelry 3111.419720798279  
1703620800 Turkey TRY Toys & games 3090.755332383303  
1703620800 Turkey TRY Health, Household & Baby Care 3092.864541436594  
1703620800 Turkey TRY Baby 3108.4507545734923  
1703620800 Turkey TRY Electronics 3092.100526659908  
1703620800 Turkey TRY Sports & outdoors 3115.856331183156  
1703620800 Turkey TRY Pet Supplies 3117.7845628814007  
1703620800 Turkey TRY Office Supplies 3117.0791667496337  
hadoop_user@hadoop-master: ~$
```

Рисунок 2 — Первые несколько строк датасета со входными данными

Необходимо построить график, отражающий почасовую выручку регионов для одной категории товаров с наибольшей суммарной выручкой в пересчете на рубли. Таким образом, задачу можно разбить на следующие подзадачи: получение данных об обмене валют; пересчет выручки для каждой совершенной покупки на рубли; построение графика (см. рисунок 3).

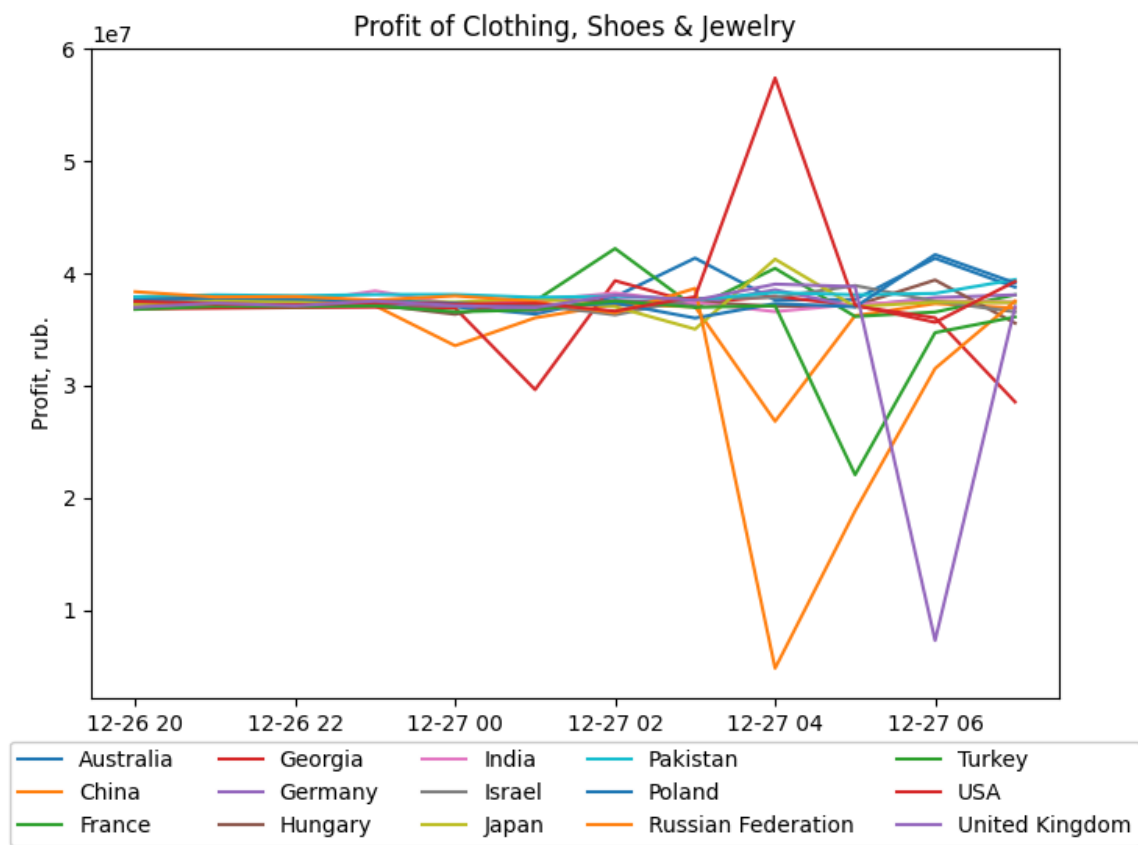


Рисунок 3 — Пример построения графа для решения задачи

Понятно, что создание кластера - не панацея для решения этой задачи (тем более на небольших данных, как будет сделано в этом проекте), но для учебных целей этого более чем достаточно.

Описание этапов работы. Здесь будут в общих чертах перечислены этапы работы по созданию кластера. Подробно описывать каждое действие с приведением команд для терминала нет смысла - для этого есть репозиторий лабораторной [7], в котором описано каждое необходимое действие и в котором приведены результаты выполнения лабораторной работы нашей командой.

Здесь же обозначим технические требования к выполнению лабораторной: устройство, на котором студент собирается выполнять лабораторную работу, должно иметь достаточные вычислительные мощности и достаточный объем памяти для одновременной работы трех виртуальных машин. Мы в работе использовали две серверные Ubuntu с дисковым пространством 10 Гб и выделенными 2 ядрами процессора и 2 Гб оперативной памяти и одну десктопную Ubuntu с 25 Гб памяти, 10 ядрами и 10 Гб оперативной памяти. Использование именно этого дистрибутива Linux не является обязательным, а выделение названного дискового пространства является обязательным, так как при

меньшем использовании памяти HDFS не сможет корректно выполнять MapReduce-задачи.

На нулевом этапе работы создаются три виртуальные машины - Master, Slave 1 и Slave 2 - которые будут планировщиком задач и рабочими узлами соответственно. Затем происходит подключение машин в одну созданную NAT-сеть и установка между ними SSH-соединения. Завершающим действием на этом этапе является установка Java на все виртуалки. В рамках этого этапа проверяются умения студента пользоваться виртуальными машинами посредством VirtualBox Manager, умение пользоваться сетевой службой NAT и терминалом. Ожидаемый результат этапа - три созданные и подключенные в одну сеть виртуальные машины.

На первом этапе происходит установка, настройка конфигураций распределенной файловой системы Hadoop, ее форматирование и запуск на созданных ранее виртуальных машинах. На этом этапе также проверяется умение пользоваться терминалом, поскольку все команды выполняются вручную, а не через какой-либо менеджер по настройке HDFS, а также умение пользоваться источниками Интернета для отладки и устранения множества потенциальных ошибок при настройке HDFS. На рисунке 4 можно увидеть ожидаемый результат выполнения этого этапа - успешный запуск распределенной файловой системы, о чем можно судить о данным о рабочих нодах в веб-интерфейсе Hadoop.

The screenshot shows the Hadoop NameNode web interface. The browser address bar indicates the URL is `localhost:9870/dfshealth.html#tab-datanode`. The page title is "In operation". Below the title, there is a "DataNode State" dropdown set to "All", a "Show" dropdown set to "25" entries, and a search bar. The main content is a table listing DataNodes. The table has columns: Node, Http Address, Last contact, Last Block Report, Used, Non DFS Used, Capacity, Blocks, Block pool used, and Version. Two DataNodes are listed, both with a green checkmark in the "Node" column, indicating they are healthy. The first node is `/default-rack/hadoop-slave1:9866 (10.0.2.9:9866)` and the second is `/default-rack/hadoop-slave2:9866 (10.0.2.10:9866)`. Both have a "Last contact" of 1s and 2s respectively, and a "Last Block Report" of 1m. The "Used" space is 24 KB, "Non DFS Used" is 6.63 GB and 6.61 GB respectively, and "Capacity" is 9.98 GB. The "Blocks" column shows 0 blocks for both. The "Block pool used" column shows 24 KB (0%) for both. The "Version" column shows 3.3.6 for both. At the bottom of the table, it says "Showing 1 to 2 of 2 entries" and there are "Previous", "1", and "Next" buttons.

Node	Http Address	Last contact	Last Block Report	Used	Non DFS Used	Capacity	Blocks	Block pool used	Version
✓/default-rack/hadoop-slave1:9866 (10.0.2.9:9866)	http://hadoop-slave1:9864	1s	1m	24 KB	6.63 GB	9.98 GB	0	24 KB (0%)	3.3.6
✓/default-rack/hadoop-slave2:9866 (10.0.2.10:9866)	http://hadoop-slave2:9864	2s	1m	24 KB	6.61 GB	9.98 GB	0	24 KB (0%)	3.3.6

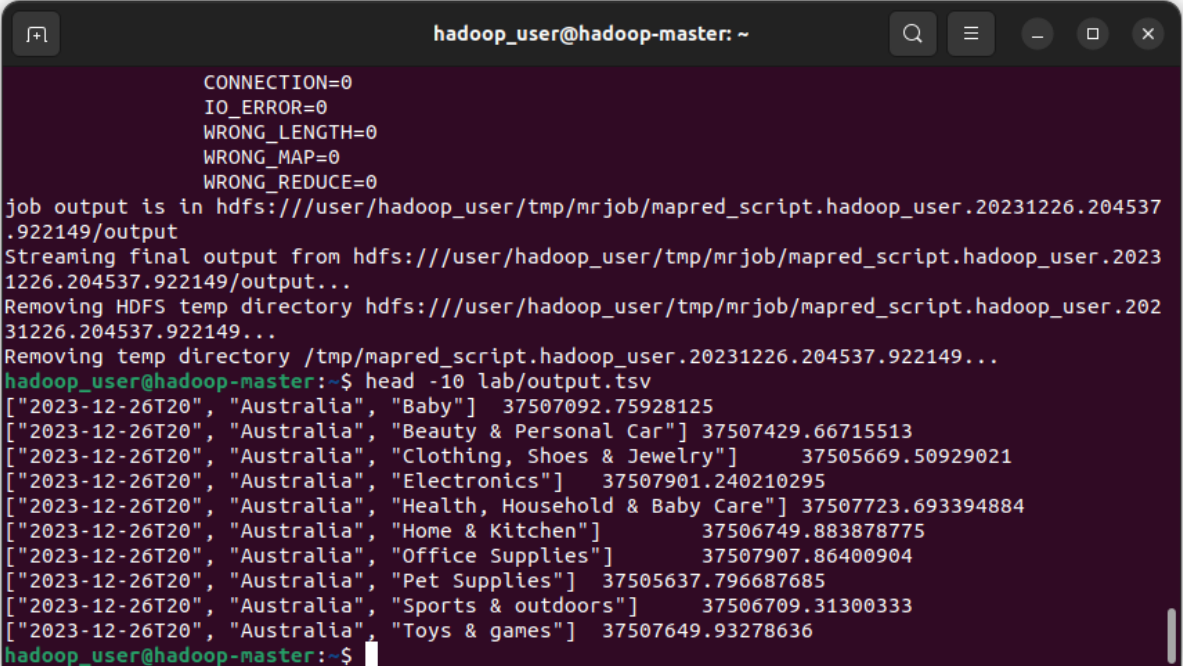
Showing 1 to 2 of 2 entries

Previous 1 Next

Рисунок 4 — Список рабочих нод кластера

На втором этапе происходит решение поставленной практической задачи (пока без Airflow). На этом этапе создаются: скрипт на Bash для получения данных о валютах; скрипт MapReduce-задачи, который пишется на python с помощью библиотеки

mrjob [8]; скрипт для визуализации данных, написанный на python с использованием библиотек pandas [9] и matplotlib [10]. На этом этапе проверяются знания написания простых скриптов на Bash [11], а также студент получает навыки написания простых MapReduce-задач и простого анализа данных. Ожидаемый результат работы этого этапа - создание описанных скриптов и последовательный их запуск, причем запуск скрипта с MapReduce-задачей должен осуществляться на созданном на предыдущем этапе кластере. на рисунке 5 можно увидеть несколько строк выходных данных, полученных после выполнения задачи на кластере, а график последнего этапа уже был показан - это рисунок 3



```
hadoop_user@hadoop-master: ~  
CONNECTION=0  
IO_ERROR=0  
WRONG_LENGTH=0  
WRONG_MAP=0  
WRONG_REDUCE=0  
job output is in hdfs:///user/hadoop_user/tmp/mrjob/mapred_script.hadoop_user.20231226.204537.922149/output  
Streaming final output from hdfs:///user/hadoop_user/tmp/mrjob/mapred_script.hadoop_user.20231226.204537.922149/output...  
Removing HDFS temp directory hdfs:///user/hadoop_user/tmp/mrjob/mapred_script.hadoop_user.20231226.204537.922149...  
Removing temp directory /tmp/mapred_script.hadoop_user.20231226.204537.922149...  
hadoop_user@hadoop-master:~$ head -10 lab/output.tsv  
["2023-12-26T20", "Australia", "Baby"] 37507092.75928125  
["2023-12-26T20", "Australia", "Beauty & Personal Care"] 37507429.66715513  
["2023-12-26T20", "Australia", "Clothing, Shoes & Jewelry"] 37505669.50929021  
["2023-12-26T20", "Australia", "Electronics"] 37507901.240210295  
["2023-12-26T20", "Australia", "Health, Household & Baby Care"] 37507723.693394884  
["2023-12-26T20", "Australia", "Home & Kitchen"] 37506749.883878775  
["2023-12-26T20", "Australia", "Office Supplies"] 37507907.86400904  
["2023-12-26T20", "Australia", "Pet Supplies"] 37505637.796687685  
["2023-12-26T20", "Australia", "Sports & outdoors"] 37506709.31300333  
["2023-12-26T20", "Australia", "Toys & games"] 37507649.93278636  
hadoop_user@hadoop-master:~$
```

Рисунок 5 — Первые строки выходных данных после применения MapReduce-скрипта на кластере

На третьем, завершающем этапе происходит оркестрация вычислений с помощью Airflow. На этом этапе на главную ноду кластера устанавливается Airflow, затем создается DAG (рисунок 6) в скрипте на python, в котором задачами являются действия, которые применялись на предыдущем этапе для решения задач.

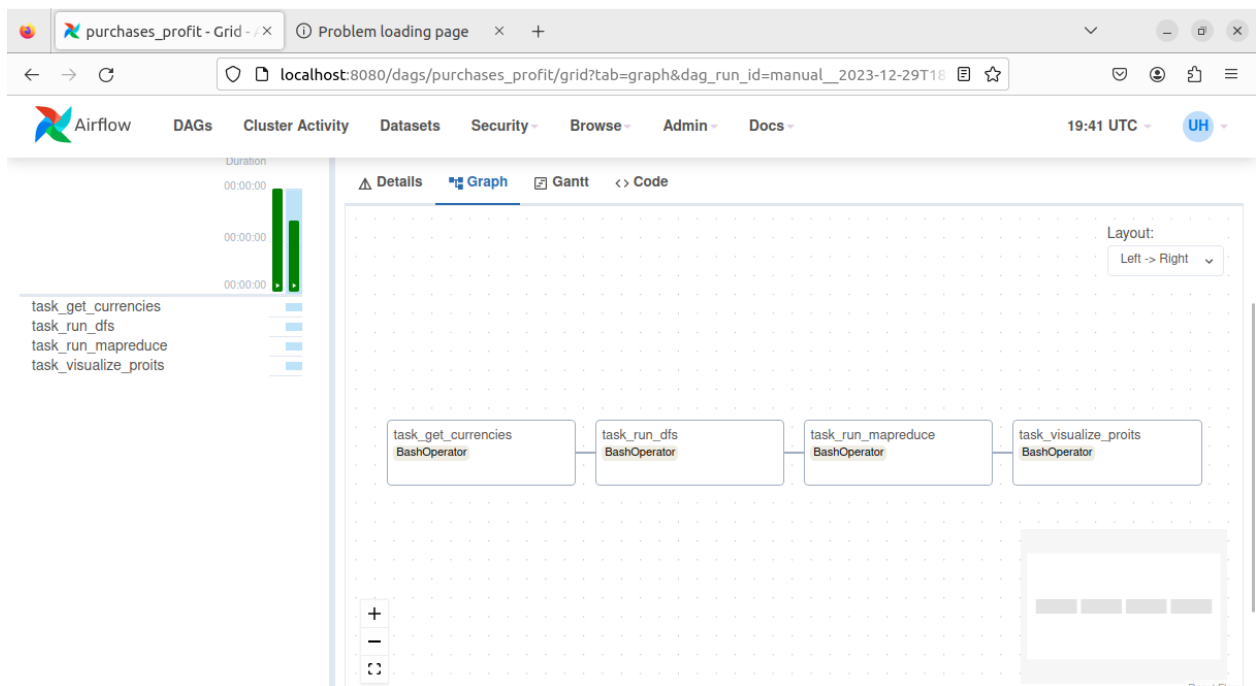


Рисунок 6 — Устройство DAG для обработки данных

После создания DAG необходимо запустить его и подождать, пока он выполнится несколько раз (см. рисунок 7). В скрипте для DAG устанавливается периодичность, с которой он должен выполняться. Мы обозначили это время за 20 минут, так как в среднем прогон данных через кластер занимает до 15 минут.

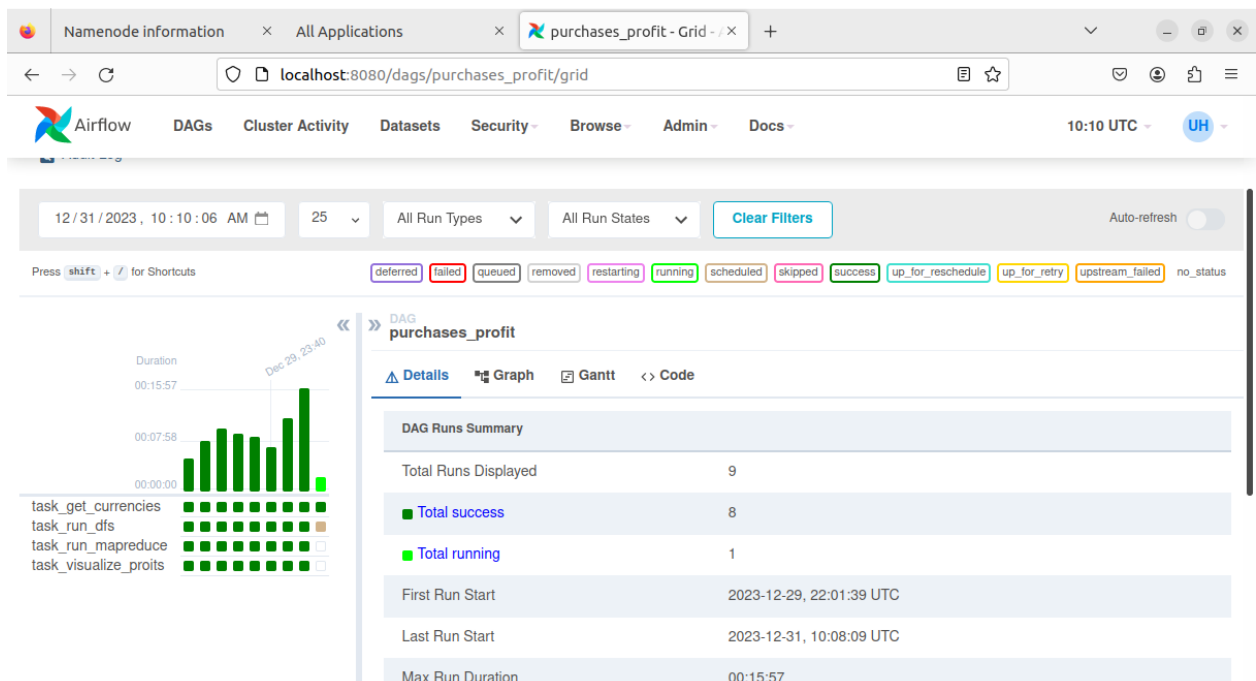


Рисунок 7 — Успешно выполненные запуски DAG для решения задачи

В рамках этого этапа проверяется умение работать с терминалом, также студент получает навыки написания простых Airflow DAG'ов. Ожидаемый результат выполнения этого этапа - созданный и запущенный несколько раз DAG, который обобщает все действия, выполняемые на предыдущем этапе. На рисунках 8 и 9 можно увидеть графики, построенные в результате 7-го и 8-го выполнения DAG'а, благодаря чему можно судить об успешности выполнения каждого этапа в нем (поскольку график отличаются, хоть и почти незаметно, поскольку за 20 минут курс валют изменился незначительно).

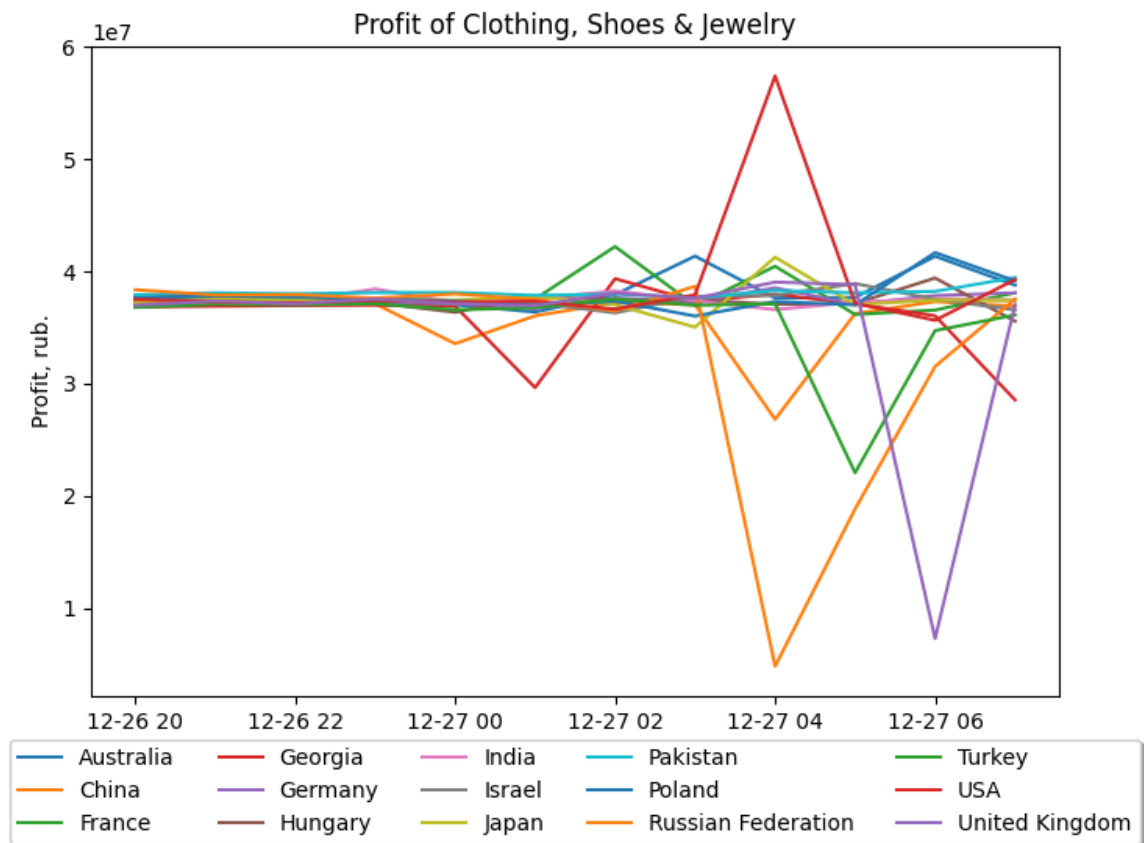


Рисунок 8 — График для 7-го выполнения DAG'а

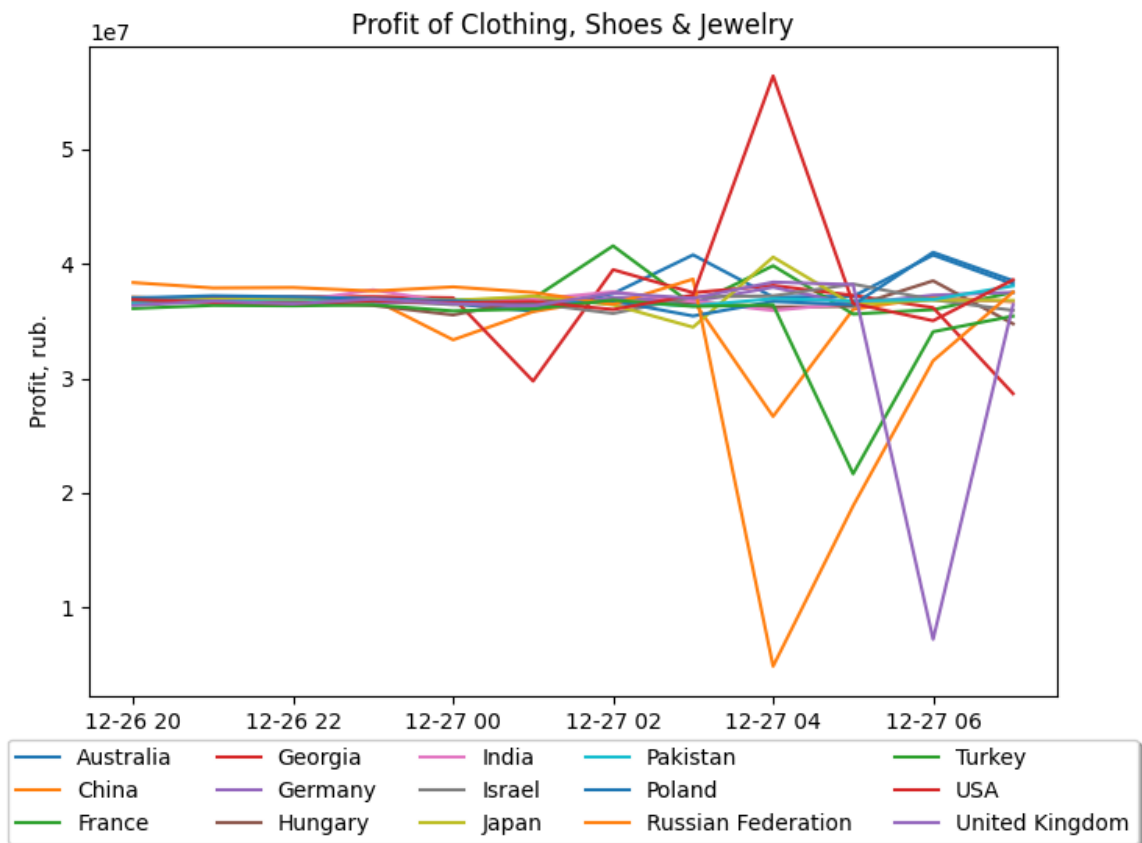


Рисунок 9 — График для 8-го выполнения DAG'а

ВЫВОДЫ.

Таким образом, описанная лабораторная работа является логическим завершением семестрового курса по информатике, поскольку она охватывает большинство изученных тем. Ожидаемый результат выполнения лабораторной работы - три созданные, настроенные и подключенные в одну сеть виртуальные машины с установленной на них распределенной файловой системой Hadoop и установленным на главный узел кластера Airflow, содержащем DAG, который выполняет поставленную практическую задачу в несколько последовательных действий. В рамках выполнения лабораторной работы проверяются знания студента в создании и настройке виртуальных машин, работе с сетевой службой NAT и протоколом SSH, работе с терминалом и Bash. В процессе выполнения работы студент получает навыки создания кластера с HDFS, создания простых MapReduce-задач и Airflow DAG'ов, а также получает небольшие практические знания в области анализа данных на python. Эти знания будут полезны в дальнейшей профессиональной деятельности. Результат работы над проектом - созданная лабораторная работа [7]. Результат ее выполнения нашей командой можно увидеть в

скриншотах, приведенных в этом отчете, в репозитории лабораторной работы и в видео-демонстрации [12].

Список литературы

1. Хабр. Big Data от А до Я. Часть 1: Принципы работы с большими данными, парадигма MapReduce [Электронный ресурс]. - URL: <https://habr.com/ru/articles/267361/> (дата обращения: 30.12.2023)
2. Tom White. Hadoop: The Definitive Guide [Электронный ресурс]. - URL: https://vk.com/doc123534285_437521202?hash=gqtMfZQhDwLBmTyFu82wuds1YNwVhpG27Po3BSA16SX&dl=z1sM1JkENQLj344pBC7C3rheYWfdIdnhGZ0crFpv1as (дата обращения: 30.12.2023)
3. Apache Hadoop 3.3.6 Documentation [Электронный ресурс]. - URL: <https://hadoop.apache.org/docs/stable/> (дата обращения: 30.12.2023)
4. Apache Hadoop YARN Documentation [Электронный ресурс]. - URL: <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html> (дата обращения: 30.12.2023)
5. Apache Airflow Documentation [Электронный ресурс]. - URL: <https://airflow.apache.org/docs/> (дата обращения: 30.12.2023)
6. Хабр. Все, что вам нужно знать об Airflow DAGs, ч.1 — Основы и расписания [Электронный ресурс]. - URL: <https://habr.com/ru/articles/682384/> (дата обращения: 30.12.2023)
7. GitHub. Репозиторий проекта [Электронный ресурс]. - URL: https://github.com/staffeev/cs_itmo_project (дата обращения: 30.12.2023)
8. mrjob v0.7.4 documentation [Электронный ресурс]. - URL: <https://mrjob.readthedocs.io/en/latest/> (дата обращения: 30.12.2023)
9. pandas 2.1.4 documentation [Электронный ресурс]. - URL: <https://pandas.pydata.org/docs/> (дата обращения: 30.12.2023)
10. Matplotlib 3.8.2 documentation [Электронный ресурс]. - URL: <https://matplotlib.org/stable/index.html> (дата обращения: 30.12.2023)
11. Mendel Cooper. Advanced Bash-Scripting Guide [Электронный ресурс]. - URL: https://se.ifmo.ru/~ad/Documentation/ABS_Guide_ru.html (дата обращения: 30.12.2023)

12. Видео-демонстрация работы кластера [Электронный ресурс]. - URL: https://drive.google.com/file/d/16BC8VS1cVjWzKh3Dgtd5pA0_921YW11D/view?usp=sharing (дата обращения: 31.12.2023)