МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Самарский национальный исследовательский

университет имени академика С.П. Королёва»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

Лабораторная работа № 1

«INTRODUCTION TO APACHE SPARK»

Выполнил студент: Ермаков Д.Д.

Гр. 6132 - 010204D

Проверил преподаватель: Попов С.Б.

Самара 2022

**ЗАДАНИЕ**

1. Изучить операции загрузки и выгрузки данных в HDFS;
2. ознакомиться с базовыми операциями Apache Spark в spark-shell;
3. создать проект по обработке данных в IDE;
4. отладить анализ данных велопарковок на локальном компьютере:
5. запустить анализ данных велопарковок на сервере.

**ХОД РАБОТЫ**

Установим виртуальную машину Hortonworks и подключаемся к ней. Копируем все необходимые для работы наборы данных и перемещаем их в файловую систему узла кластера (Рисунок 1).

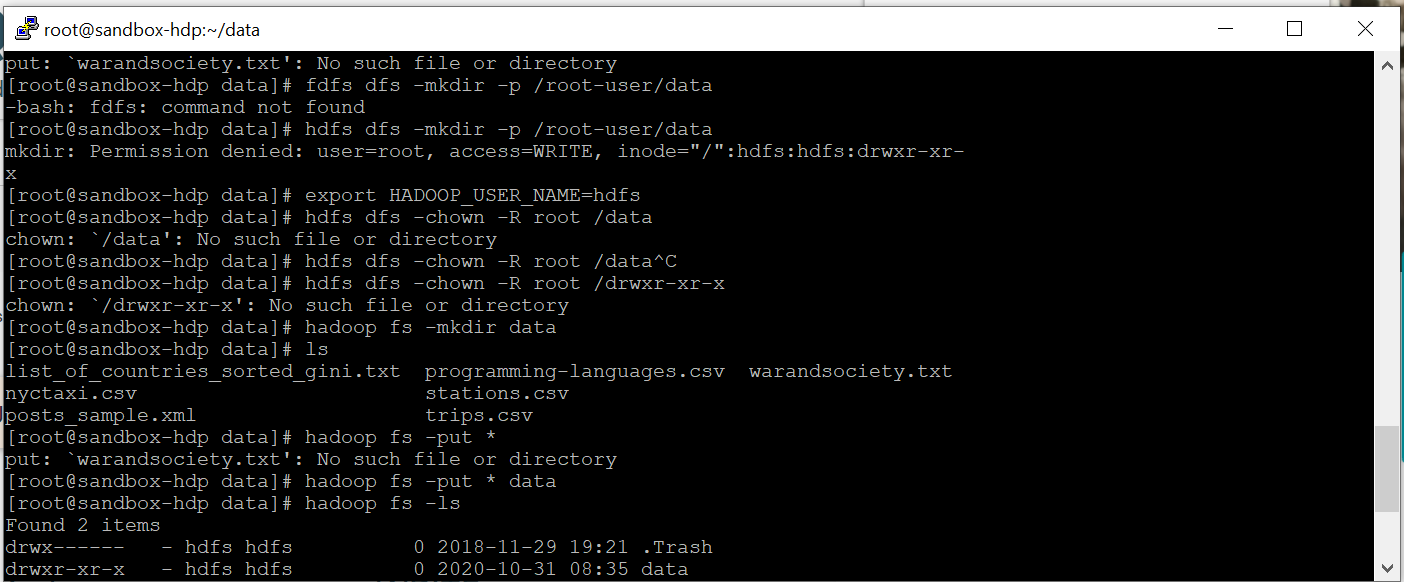


Рисунок 1 – Перемещение данных в директорию

Запускаем spark-shell и открываем веб интерфейс (Рисунок 2 и 3).

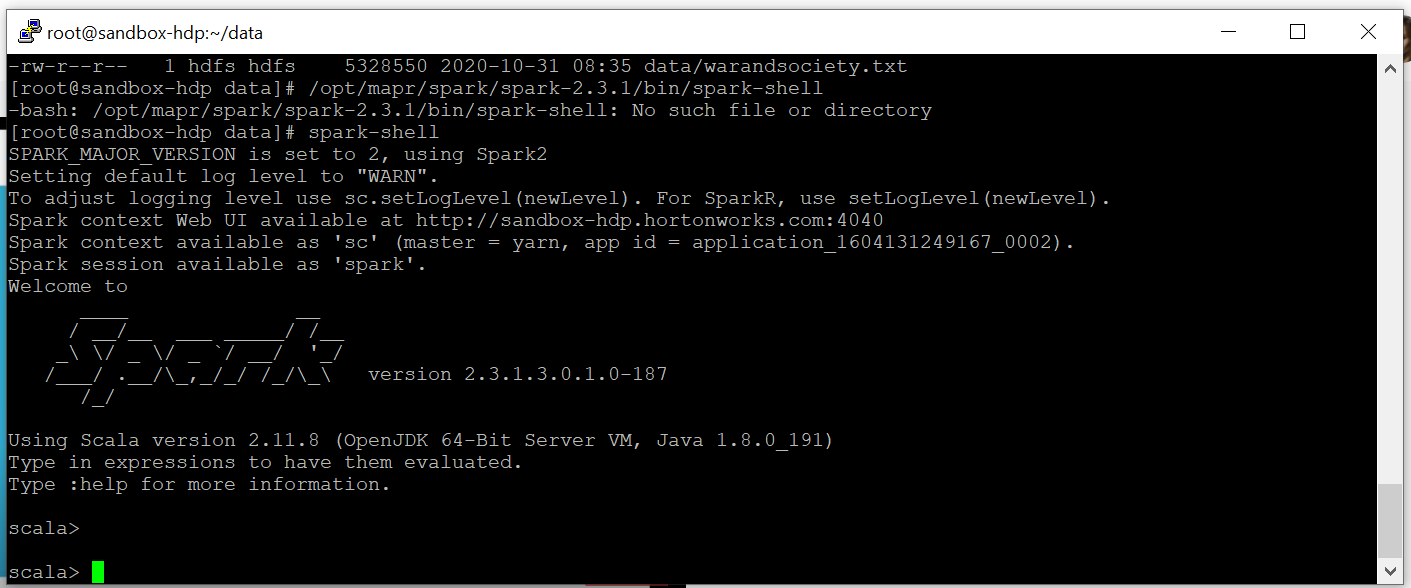


Рисунок 2 – Запуск spark-shell

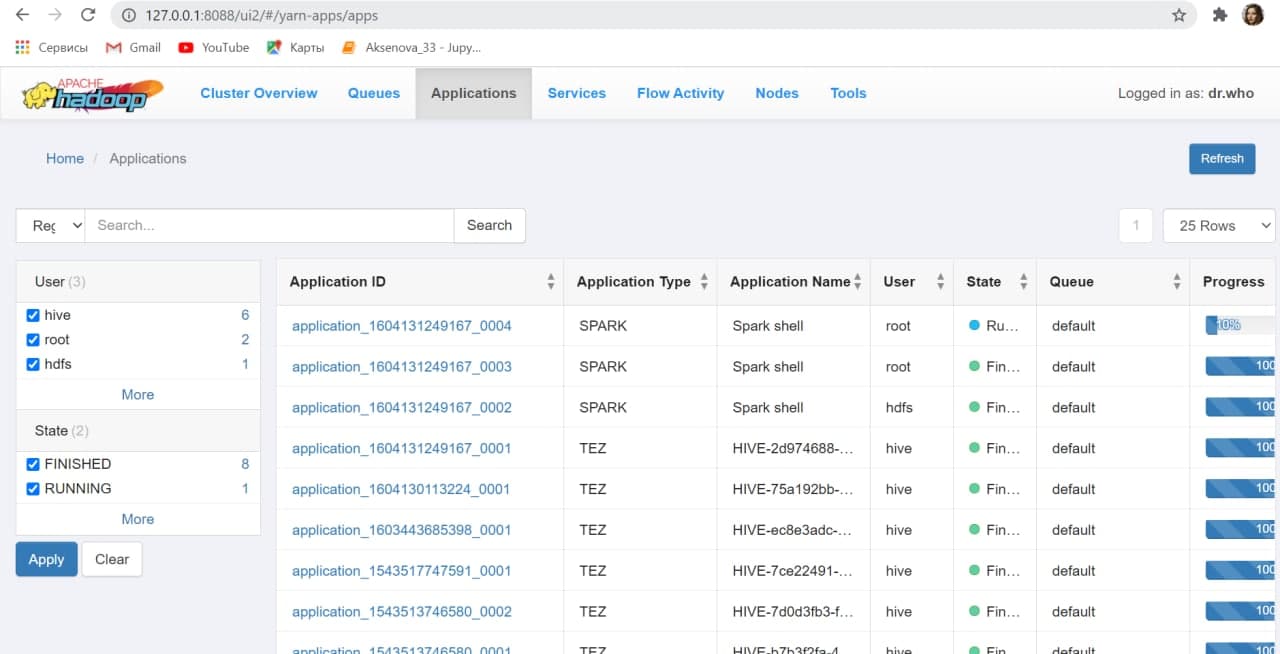


Рисунок 3 – Web-страница spark-shell

Создаем RDD для текстового файла warandpeace.txt и считаем первые 10 строк файла (Рисунок 4 и 5).

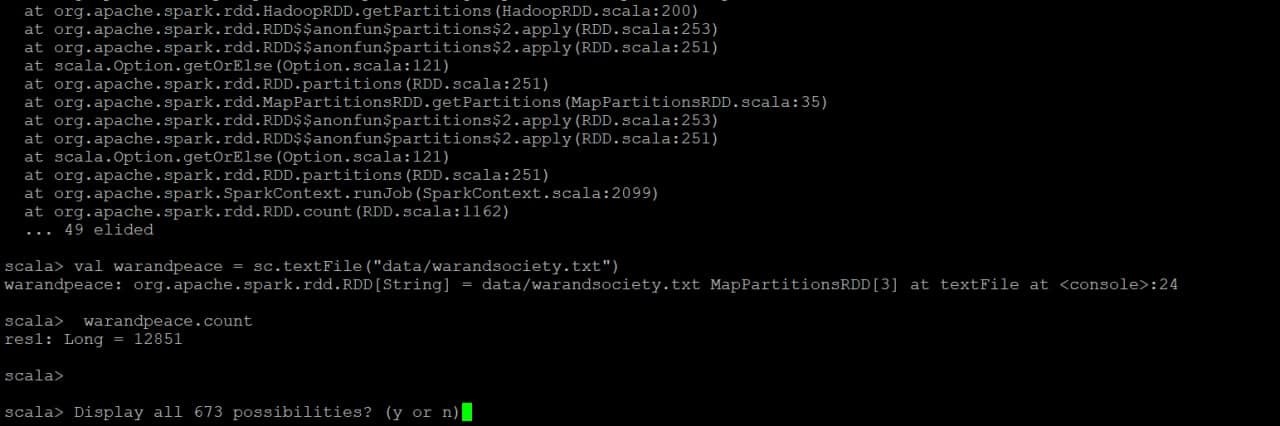


Рисунок 4 – Создание RDD

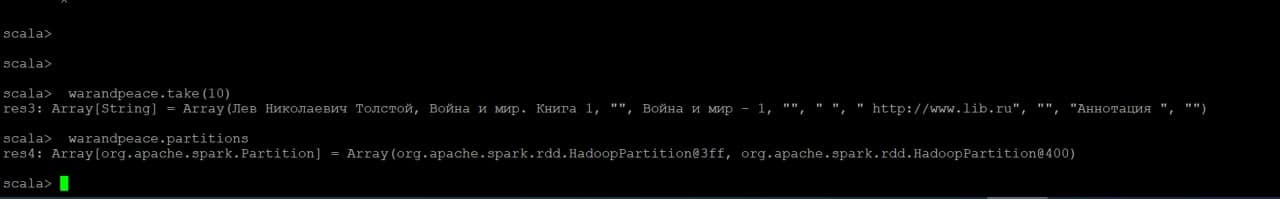


Рисунок 5 – Считывание строк RDD

Найдем строки, в которых содержится слово "война". Подсчитаем количество строк несколько раз и замерим время работы программы. При повторном выполнении команды count, замечаем небольшое ускорение работы программы за счет кэширования (Рисунок 6 и 7).

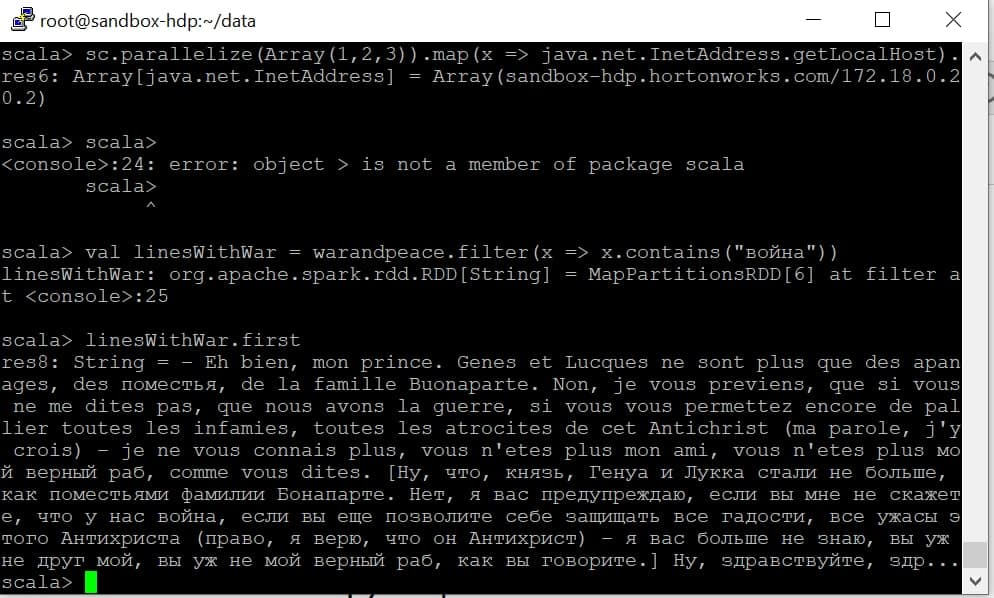


Рисунок 6 – Строка со словом «война»

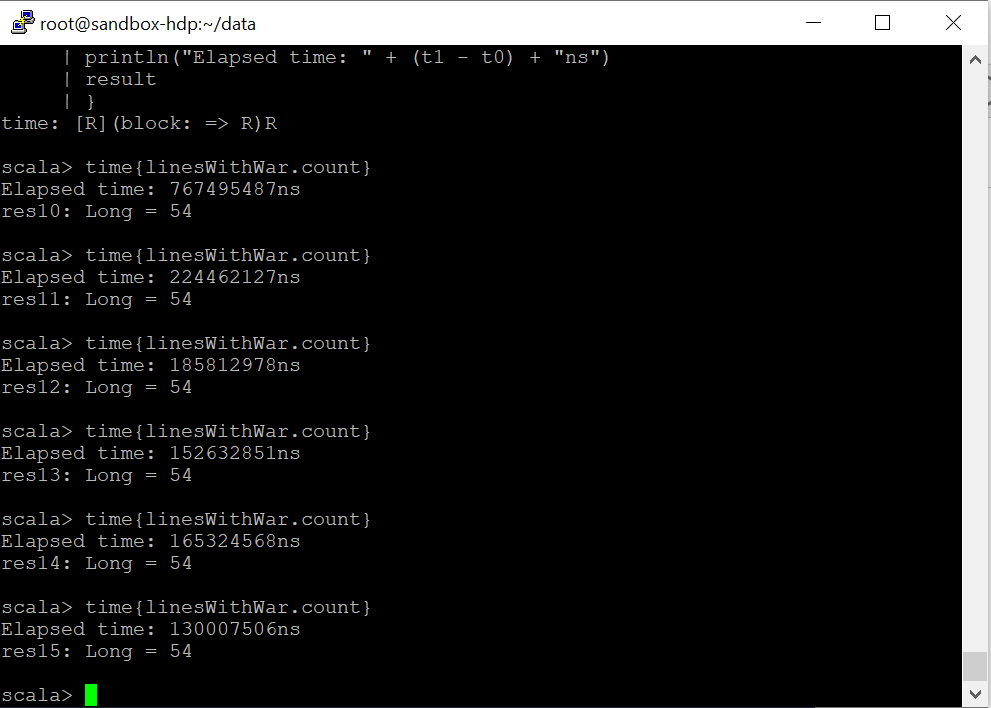


Рисунок 7 – Ускорение за счет кэширования

Найдем гистограмму слов (Рисунок 8).

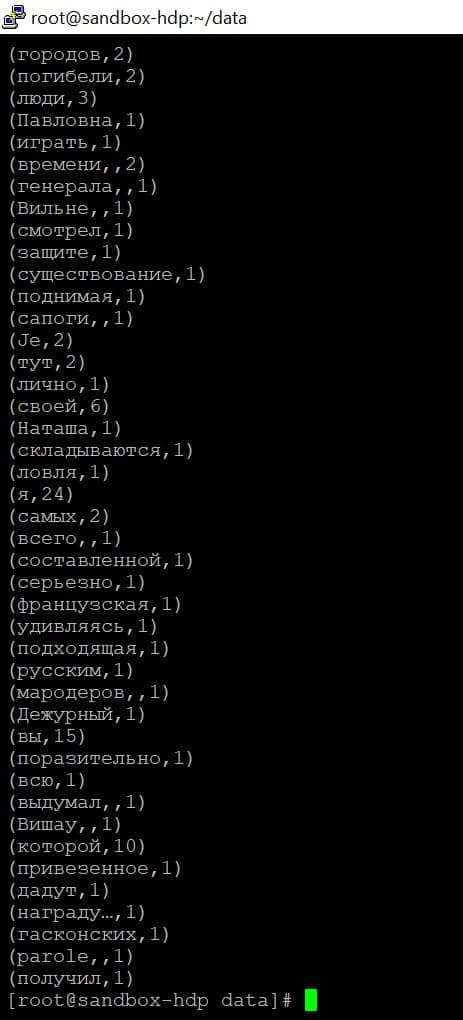


Рисунок 8 – Гистограмма слов

Улучшаем процедуру, убирая лишние символы и трансформируя все слова в нижний регистр, используя регулярные выражения (Рисунок 9 и 10).

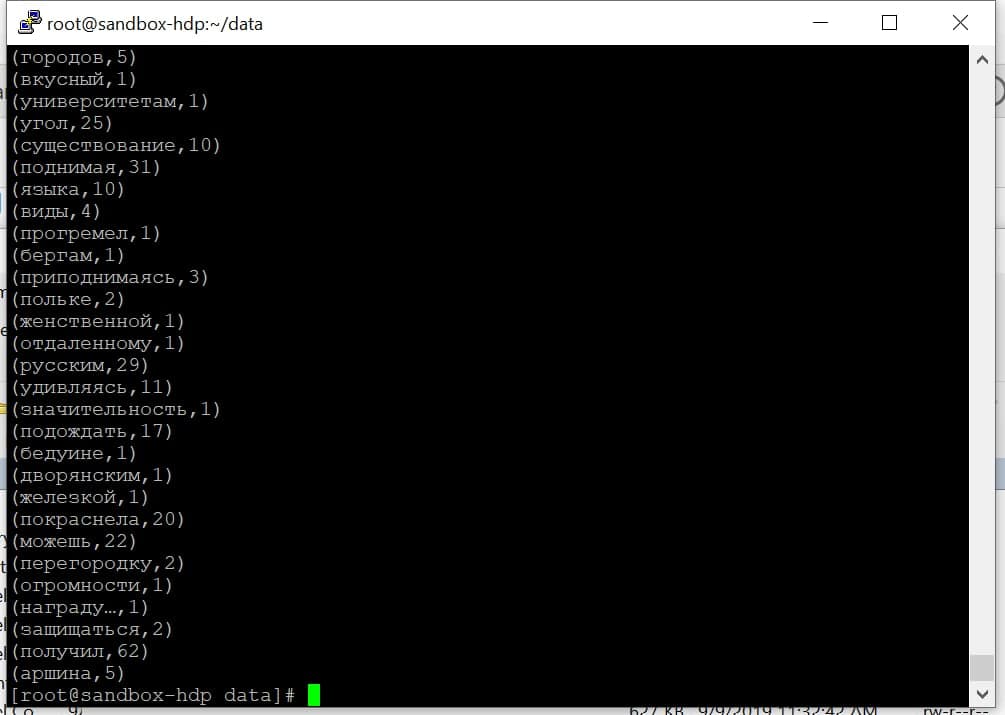


Рисунок 9 –Гистограмма слов без лишних символов

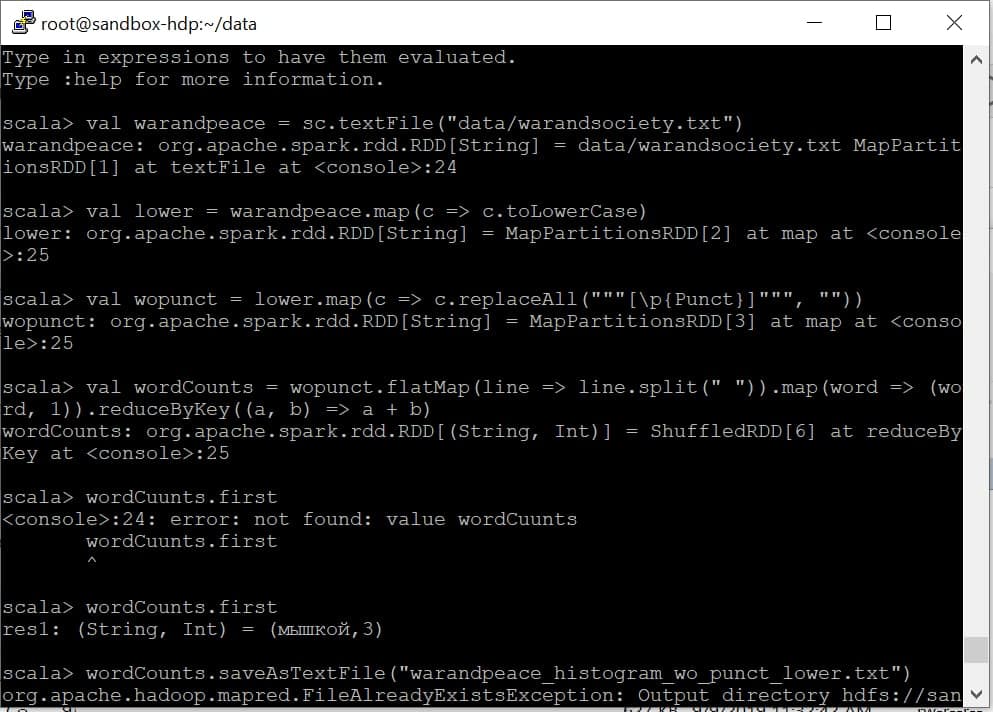


Рисунок 10 –Улучшение гистограммы

Инициализируем два множества и найдем их объединение. Убирем из него дубликаты с помощью функции distinct и найдем пересечение с разностью множеств (Рисунок 11).

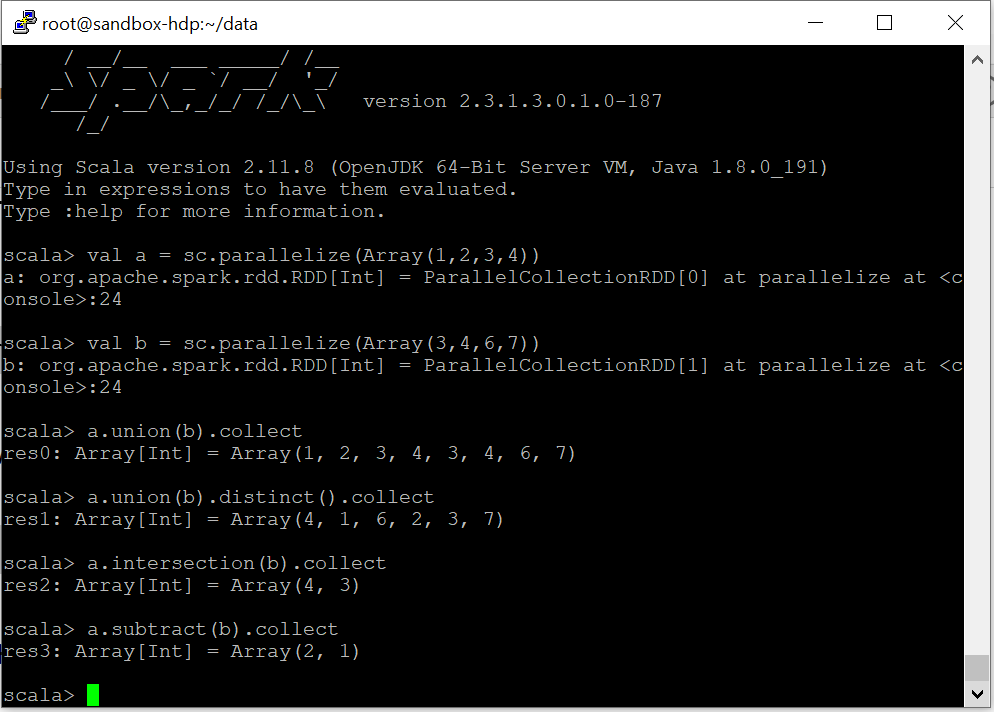


Рисунок 11 – Операции над множествами

Создадим широковещательную переменную (Рисунок 12).

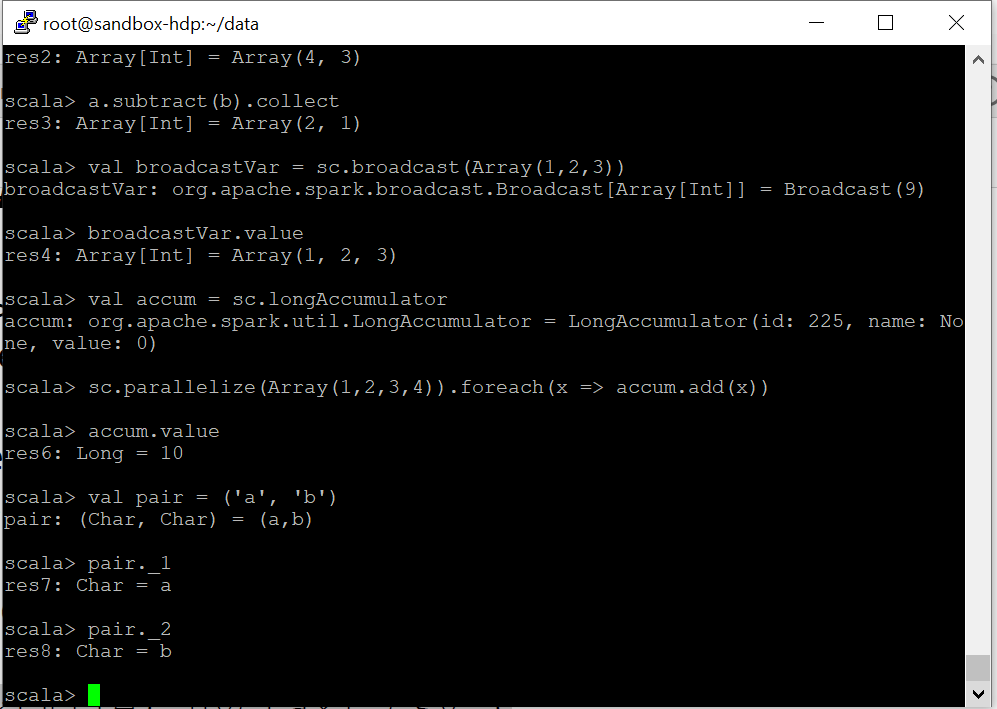


Рисунок 12 – Широковещательная переменная

Создадим аккумулирующую переменную (Рисунок 13).

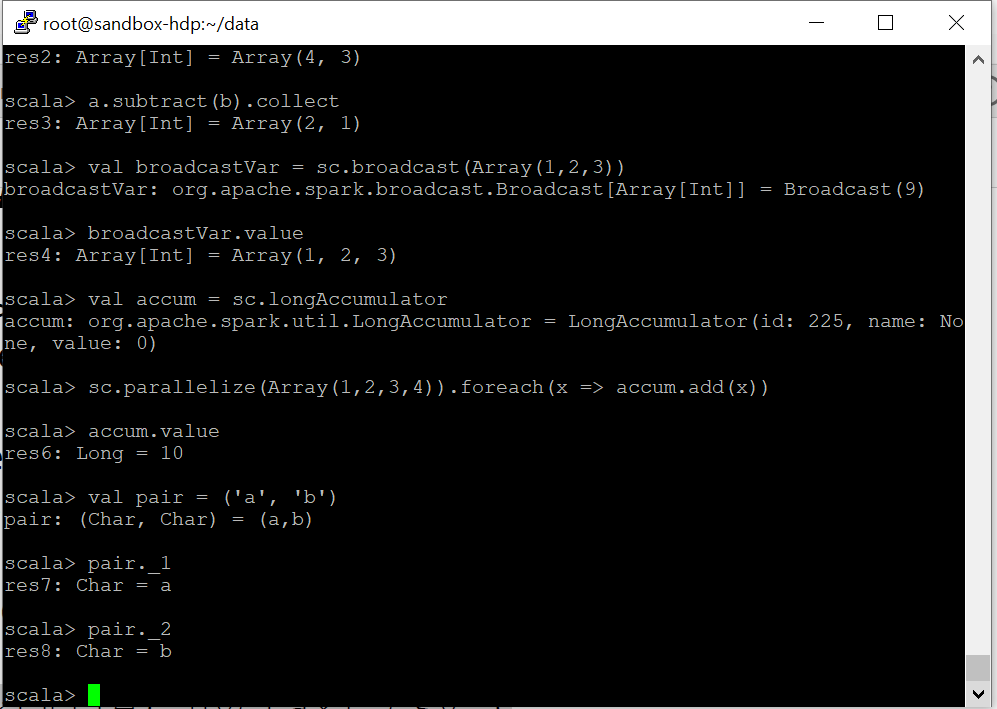


Рисунок 13 – Аккумулирующая переменная

Создадим пары ключ-значение из двух букв (Рисунок 14).

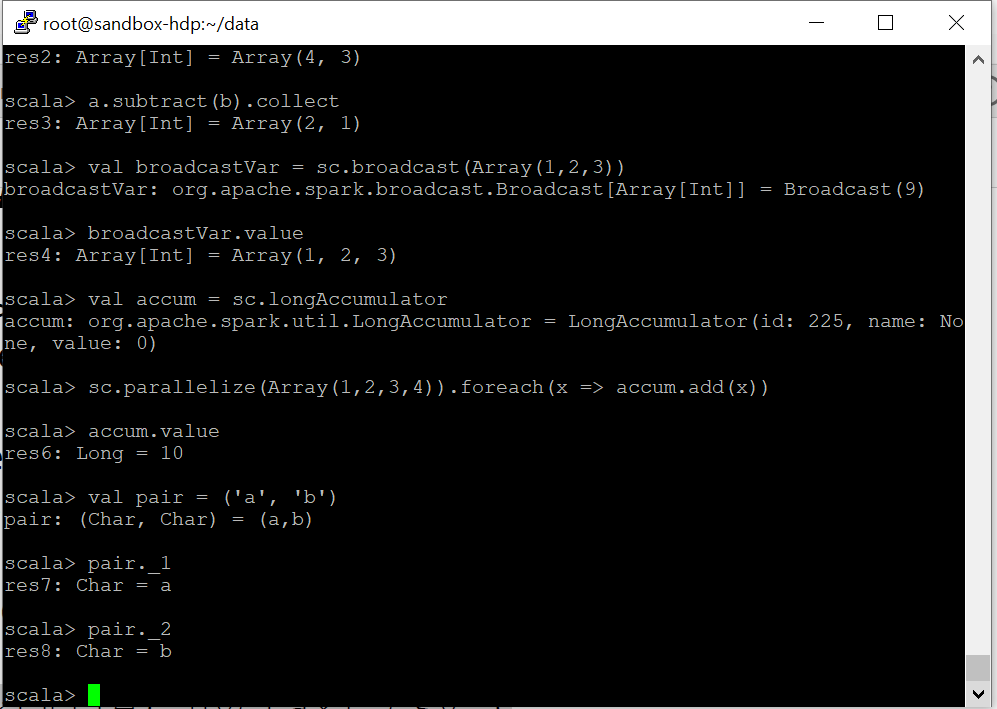


Рисунок 14 – Пары ключ-значение

Проанализируем данные о поездках такси в Нью-Йорке и найдем первые 10 номеров такси, которые совершили наибольшее количество поездок (Рисунок 15).

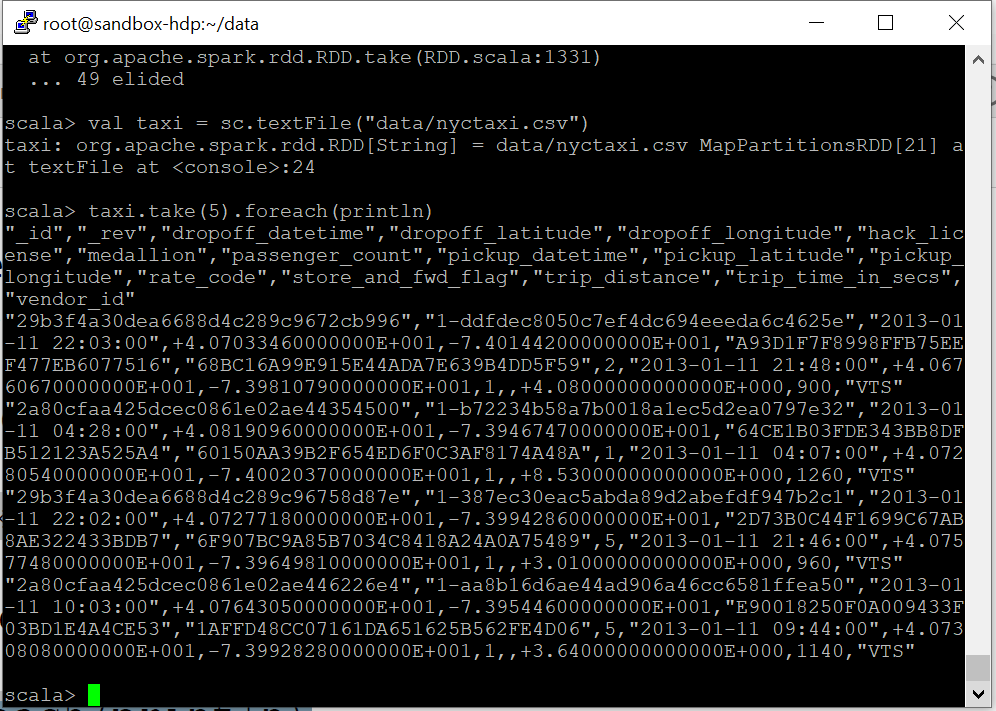


Рисунок 15 – Такси совершившие наибольшее количество поездок

Найдем количество поездок каждого номера такси и выведем полученные результаты в отсортированном виде (Рисунок 16).

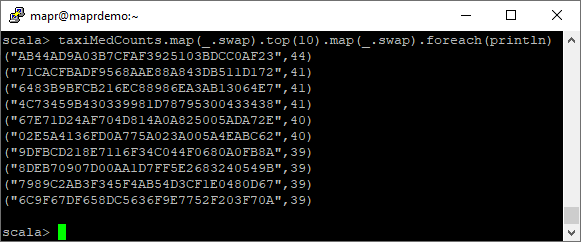


Рисунок 16 – Количество поездок каждого номера такси

Найдем общее количество номеров такси, несколько раз предварительно объявив RDD taxiCounts как сохраняемую в кэше. Время выполнения операции уменьшается за счёт кэширования (Рисунок 17).

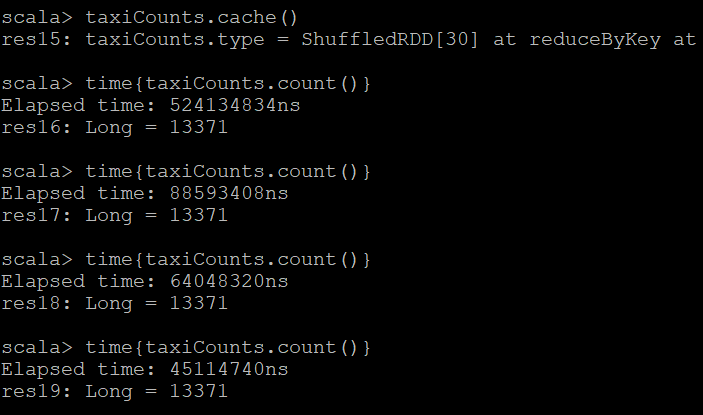


Рисунок 17 – Время подсчета общего числа такси

В ходе выполнения работы был сделан проект в IntelliJ IDEA на языке Scala. Был произведен анализ данных велопарковок и поездок на велосипедах. Был произведен запуск проекта на Hortonworks.

Выводы данных по результатам выполнения работ находятся в файле README.md в репозитории https://github.com/ShadowOfDen/BigData.

**ЗАКЛЮЧЕНИЕ**

В результате лабораторной работы были изучены операции загрузки и выгрузки данных в HDFS, базовые операции Apache Spark в spark-shell. В IntelliJ IDEA был создан проект по обработке данных на локальном компьютере и были выполнены задания и проведен анализ данных. А сам проект был запущен на виртуальной машине Hortonworks.