Работа с MapReduce

1. Стандартный пример - подсчёт числа слов

Подготовьте файл `Работа с MapReduce - отчёт.docx`. Помещайте в него выполняемые команды и вставляйте скриншоты выводимых на экран результатов.

После выполнения задания "Работа с HDFS" у Вас уже должен быть скачан и распакован архив `bigd_course_hadoop.zip`. Если нет - скачайте и распакуйте его.

Перейдите в консоли в папку `bigd_course_hadoop` и, используя имеющийся в ней конфигурационный файл, поднимите кластер Hadoop и дождитесь, когда все его узлы перейдут в статус `healthy`.

Если по окончании выполнения задания по работе с HDFS Вы удалили тома кластера Hadoop, заново скопируйте содержимое папки `files` в файловую систему узла имён, в папку `/home/user`.

Перейдите в файловую систему контейнера узла имён.

```
docker exec -it namenode bash
```

Распакуйте файлы `eng_wiki.txt.gz` и `words.txt.gz` следующими командами:

```
gzip -d /home/user/eng_wiki.txt.gz
gzip -d /home/user/words.txt.gz
```

Скопируйте файлы `eng_wiki.txt` и `words.txt` в папку `/user/data` файловой системы HDFS. При необходимости создайте отсутствующие папки.

В состав Hadoop входят несколько тестовых приложений MapReduce. Одно из них - подсчёт слов в текстовом файл.

Запустите приложение подсчёта слов для подсчёта слов в файле `words.txt`. Результат работы будет помещён в папку `/user/output/jar_wordcount` в HDFS:

```
yarn jar \
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar \
wordcount \
/user/data/words.txt \
/user/output/jar_wordcount
```

Здесь звёздочка обозначает номер версии имеющегося файла. Когда будете набирать эту команду, после `hadoop-mapreduce-examples-` нажмите на клавиатуре клавишу Tab и правильные числа будут подставлены автоматически.

Обратите внимание на порядок аргументов. Сначала указывается исполняемые jar-файл, затем название выполняемой задачи `wordcount`, обрабатываемый файл `/user/data/words.txt` и в конце - имя папки в HDFS для сохранения результата `/user/output/jar_wordcount`.

Замечание: если папка для результатов работы (в нашем примере это `/user/output/jar_wordcount`) уже существует в HDFS, то задача будет остановлено с сообщением об ошибке. В этом случае нужно удалить папку `jar_wordcount`.

Откройте веб-интерфейс YARN по адресу http://localhost:8088) и на главной странице "All Applications" найдите запущенное приложение. Дождитесь его завершения и поместите в отчёт следующую информацию:

- пользователь, от имени которого было запущено приложения
- имя приложения
- его тип
- время запуска приложения
- время окончания исполнения
- статус завершения

Результат работы приложения находится в папке HDFS `/output/words`, которая была указа при запуске приложения.

Выведите на экран находящиеся в ней файлы и поместите их имена в отчёт. Напишите в отчёте, что в этой папке является признаком корректного завершения приложения.

Выведите на экран содержание файла, в который помещены подсчитанные количества слов. Поместите это в отчёт.

2. Подсчёт числа слов на Python

Находясь в локальной файловой системе узла имён, перейдите в папку `/home/mapreduce/wordcount` которая подмонтирована из файловой системы хоста и содержит два файла Python, `mapper.py` и `reducer.py`. Проверьте это, просмотрев содержание этой папки.

```
cd /home/mapreduce/wordcount
ls
```

Использование Python для MapReduce задач осуществляется с применением Streaming API Hadoop. Для этого требуется подготовить отдельные программы на Python для операций Map и Reduce.

Основной принцип работы этих программ - чтение входных данных из стандартного потока ввода `stdin` и передача результата своей работы в стандартный поток вывода `stdout`.

Подсчёт числа слов в рассматриваемом примере делается следующим образом.

- Mapper (`mapper.py`) читает строки из `stdin`, разбивает текст статьи на слова и выводит в `stdout` пары ключ-значение. В нашем случае это `(слово, 1)`.
- Reducer (`reducer.py`) суммирует счетчики для каждого слова и выводит в `stdout` пары `(слово, сумма)`.

Откройте эти файлы и изучите их код. Это можно сделать используя приложения хоста.

Другой вариант просмотра - запустить при помощи консольной команды `mc` в файловой системе узла имён двухпанельный файловый менеджер Midnight Commander . Чтобы просмотреть файл при помощи MC, нужно перейти к нужному файлу, клавишами со стрелками навести на него строку выделения и нажать клавишу F3 (просмотр) или F4 (редактирование). Выход из MC - клавиша F10.

Важно помнить, что после операции Map YARN выполняет операцию Shuffling которая сводится к упорядочиванию по ключу данных, полученных от Mapper. Это значит, что Reducer получает одинаковые слова подряд, что упрощает алгоритм его работы.

Изучите команду запуска MapRedcue задачи, которая приведена ниже. Обратите внимание как заданы имена программ Mapper и Reducer - путь к ним не указан. Это означает, что программы будут взяты из текущей папки. Чтобы это сработало, перед запуском нужно перейти в папку локальной файловой системы узла имён в котором лежат эти файлы (альтернатива - запускать MapReduce из произвольной папки, но указывать полные пути к файлам программ). Также обратите внимание на то, как задан обрабатываемый файл и папка, в которую будут помещены результаты работы.

Еще один важный момент - каждый из исполняемых файлов указан дважды - в аргументе `-mapper` или `-reducer` и в аргументе `-files`. Это нужно из-за принципа локальности данных - код доставляется к месту хранения данных. Аргумент `-files` содержит перечень файлов, которые потребуются при выполнении задачи и поэтому должны быть доставлены на узлы данных для исполнения.

Выполните MapRedcue задачу.

```
yarn jar \
$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
-files mapper.py,reducer.py \
-mapper "python3 mapper.py" \
-reducer "python3 reducer.py" \
-input /user/data/words.txt \
-output /user/output/py_wordcount
```

(Не забудьте вместо звёздочки вписать номер версии файла - нажмите клавишу Tab, когда будете набирать эту команду.)

Результат работы приложения находится в папке HDFS `/output/words`, которая была указа при запуске приложения.

Выведите на экран находящиеся в ней файлы и поместите их имена в отчёт. Напишите в отчёте, что в этой папке является признаком корректного завершения приложения.

Выведите на экран содержание файла, в который помещены подсчитанные количества слов. Поместите это в отчёт. Сравните с результатом предыдущей задачи.

3. Создание скрипта запуска задачи

Мы собираемся создать скрипт командной оболочки Linux, с которой мы работаем, когда попадаем в консоль узла имён. В Linux используются различные командные оболочки. На узлах нашего кластера используется самая «каноническая» оболочка `bash`.

Создайте в папке `mapreduce/wordcount` пустой текстовый файл с именем `run.sh`. Это можно сделать, работая на хосте при помощи простого текстового редактора.

Альтернативный вариант - находясь в файловой системе узла имён перейдите в папку `/home/mapreduce/wordcount` и выполните в консоли команду Linux `touch`, которая создаёт пустой файл, а затем убедитесь при помощи `ls` что файл действительно создан:

```
touch run.sh
ls
```

Далее можно продолжить работу с созданным файлом `run.sh` в простом текстовом редакторе хоста или в консоли узла имён запустить MC и клавишей F4 открыть файл на редактирование.

В самой первой строке созданного файла поместите специальный комментарий, который будет сообщать Linux, что это скрипт, который нужно передать на обработку командной оболочке `bash`:

```
#!/bin/bash
```

После этой строки добавьте команду удаления папки с результатами выполнения задачи - в противном случае при повторных запусках задачи будет сообщение об ошибке:

```
hdfs dfs -rm -R /user/output/py_wordcount
```

Затем скопируйте команду запуска MapReduce задачи на Python, которую Вы выполняли выше. Файл в целом должен выглядеть так:

```
#!/bin/bash

hdfs dfs -rm -R /user/output/py_wordcount

yarn jar \
$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-*.jar \
-files mapper.py,reducer.py \
-mapper "python3 mapper.py" \
-reducer "python3 reducer.py" \
-input /user/data/words.txt \
-output /user/output/py_wordcount
```

Сохраните сделанные изменения.

Находясь в консоли узла имён, в папке `/home/mapreduce/wordcount`, где находится файл `run.sh`, выполните команду Linux, которая изменяет режим доступа и использования созданного файла. Во-первых, нужно сделать этот файл исполняемым, и во-вторых, нужно разрешить всем (владельцу, группе и другим) читать и изменять этот файл. Если это не сделать, то доступ к этому файлу будет ограничен, в том числе на машине хоста.

```
chmod a+rwx run.sh
```

Здесь `chmod` - команда Linux, которая изменяет режим доступа к файлу, `a+rwx` - опция, которая прочитывается как «Всем (`a`) добавить (`+`) разрешение читать (`r`), писать (`w`) и исполнять (`x`) указанный файл».

Теперь вызовите на исполнение созданный скрипт, набрав *путь к нему* и имя, и убедитесь, что MapReduce задача корректно выполнятся.

```
./run.sh
```

Обратите внимание, что указание пути к скрипту обязательно. В противном случае он не будет запущен. Если команда запуска выполнеятся из папки, в которой находится скрипт, путь выглядит так `./` (символ `.` обозначает текущую папку).

4. Подсчёт слов в большом тексте. Удаление мусора

Файл `eng_wiki.txt` содержит статьи из англоязычной Википедии с удалённой разметкой. Познакомьтесь с содержанием этого файла. Он файл находится в локальной файловой системе узла имён по адресу `/home/user/`, а также должен быть размещён в HDFS по адресу `/user/data`.

Выведите на экран первые 20 строчек этого файла из локальной файловой системы:

```
head -20 /home/user/eng_wiki.txt
```

Создайте папку `mapreduce/wordcount_filtered` и скопируйте туда все файлы из `mapreduce/wordcount`.

Измените скрипт запуска так, чтобы он обрабатывал файл `/user/data/eng_wiki.txt`, а результаты записывались в папку HDFS `/user/output/py_wordcount_filtered`.

При написании скрипа примните во внимание версию Python, установленного на кластере. Чтобы узнать версию, наберите в консоли команду `python3`. Скорее всего это будет не самая последняя версия программы и некоторые функции могут оказаться недоступными. В частности, не будут работать f-строки.

Запустите скрипт, дождитесь окончания его работы.

Выведите на экран начало полученного в результате файла с подсчитанными словами. Скопируйте несколько строк в отчёт. Вы увидите, что использованные программы подсчёта слов оставляют очень много мусора - учитываются знаки препинания и цифры.

Перепишите Mapper так, чтобы читаемый текст сначала приводился бы к нижнему регистру, а затем из него извлекались бы слова по следующим признакам:

- в слове могут быть только буквы или апостроф,
- апостроф не может стоять в начале слова,
- в слове не может быть два или более апострофов,
- слово должно иметь длину более двух символов,
- слово не должно принадлежать списку стоп-слов.

Стоп-слова, или «шумовые слова», - это слова и фразы (например, предлоги, частицы, местоимения, вводные слова), которые не несут смысловой нагрузки и делают текст перегруженным, избыточным и трудным для восприятия.

Множество стоп-слов возьмите отсюда:

```
stop_words = {
    'the', 'a', 'an', 'and', 'or', 'but', 'in', 'on', 'at', 'to', 'for',
    'of', 'with', 'by', 'from', 'up', 'about', 'into', 'through', 'during',
    'before', 'after', 'above', 'below', 'between', 'among', 'is', 'are',
    'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having',
    'do', 'does', 'did', 'doing', 'would', 'could', 'should', 'may', 'might',
    'must', 'can', 'shall', 'will', 'this', 'that', 'these', 'those', 'i',
    'you', 'he', 'she', 'it', 'we', 'they', 'me', 'him', 'her', 'us', 'them'}
```

Перед тем как запускать выполнение задачи на кластере, протестируйте её в локальной файловой системе узла имён. Для этого сначала скопируйте небольшое количество строк из файла `eng_wiki.txt` в файл `eng_wiki_small.txt`, который расположите в той же папке, что и разрабатываемые Mapper и Reducer. Используйте следующую команду Linux:

```
head -20 /home/user/eng_wiki.txt >
/home/mapreduce/wordcount_filtered/eng_wiki_small.txt
```

Обратите внимание на то, как работает эта команда. Сначала `head -20 /home/user/eng_wiki.txt` читает первые 20 строк из файла `eng_wiki.txt`. По умолчанию она выводит прочитанную информацию на стандартное устройство вывода - экран. Но мы используем команду перенаправления вывода: знак больше `>` говорит, что вывод команды нужно перенаправить в файл, указанный после неё.

Затем протестируйте свои Mapper и Reducer создав следующий трубопровод (pipe) в Linux:

```
cat eng_wiki_small.txt | python3 mapper.py | sort | python3 reducer.py
```

Эта команда состоит из четырёх частей. Первая часть `cat eng_wiki_small.txt` выводит на стандартное устройство вывода содержание файла `eng_wiki_small.txt`. Далее оператор вертикальной черты `|` связывает стандартный поток вывода команды слева и стандартный ввод команды справа от него. После идёт команда выполнения кода Mapper, которая читает стандартный ввод, обрабатывает его и отправляет свои данные на стандартный вывод. Следующий элемент - команда `sort`. Она моделирует операцию Shuffling, выполняемую на кластере Hadoop. Наконец, последнее звено трубопровода - операция Reducer, которая печатает свой вывод на экране.

Когда отладка будет закончена, выполните задачу на кластере, выведите на экран начало файла результатов и поместите в отчёт несколько строк из него.

5. Распределение слов по длине

Сделайте копию файлов из `mapreduce/wordcount_filtered` в новой папке `mapreduce/wordlengths`.

Измените скрипт запуска так, чтобы результаты записывались в папку HDFS `/user/output/py_wordlengths`.

Измените код программ, выполняющих MapReduce задачу таким образом, чтобы выполнялся подсчёт количества слов одинаковой длины. Файл результатов в левой колонке должен содержать длину слова, а в правой - число слов такой длины. Условия отбора слов оставьте прежними.

Записи должны быть упорядочены по возрастанию длины. При этом запрещается накапливать данные и выполнять сортировку в Reducer. Правильная сортировка должна быть обеспечена операцией Shuffling.

6. Самые частые слова

Сделайте копию файлов из `mapreduce/wordcount_filtered` в новой папке `mapreduce/topwords`.

Измените скрипт запуска так, чтобы результаты записывались в папку HDFS `/user/output/py_topwords`.

Измените код программ, выполняющих MapReduce задачу таким образом, чтобы они вычисляли пять наиболее часто встречающихся слов и частоты их появления. Вывод должен быть упорядочен по убыванию частоты.

После того как выполнение задания окончено, выйдите из консоли узла имён и остановите кластер

```
docker compose down
```

Для очистки долговременной памяти своего компьютера, удалите тома файловых систем узлов кластера командой ниже.

```
docker volume prune --all
```

ВАЖНО: все локальные файловые системы узлов кластера, а также все файлы в HDFS будут удалены! Не делайте этого, пока не отчитаетесь по заданию преподавателю и не закончите выполнение всех заданий по работе с кластером Hadoop.