**Задание к лабораторной работе №1 «Apache Hadoop — MapReduce» по курсу «Распределенные и облачные вычисления, параллельное программирование».**

Преподаватель: Данилов Игорь Геннадьевич, к.т.н., ассистент кафедры МОП ЭВМ; контакты: email — [igdanilov@sfedu.ru](mailto:igdanilov@sfedu.ru), вконтакте – id1274906.

Дата обновления документа: 13.02.2017

**Общие сведения**

Лабораторная работа выполняется и сдается бригадой, состоящей из одного-двух человек. Для выполнения лабораторной работы необходимо скачать и импортировать виртуальную машину Hortonworks Sandbox (<http://hortonworks.com/downloads/#sandbox>).

Полезные команды:

1. создание директории HDFS: hadoop fs -mkdir -p /myparentdir/mydir;
2. загрузка файлов из локальной директории в HDFS: hadoop fs -put ./localdir/\* /myparentdir/mydir;
3. просмотр содержимого директории HDFS: hadoop fs -ls /myparentdir/mydir;
4. просмотр текстового файла в директории HDFS: hadoop fs -cat /myparentdir/mydir/file.txt;
5. загрузка файлов из HDFS в локальную директорию: hadoop fs -get /myparentdir/mydir/\* ./localdir/;
6. удаление содержимого директории HDFS: hadoop fs -rm -r -f /myparentdir/mydir.

Компиляция исходников:

# mkdir build

# export HADOOP\_CLASSPATH=$JAVA\_HOME/lib/tools.jar

# hadoop com.sun.tools.javac.Main MyWork.java -d build

# jar -cvf WordCount.jar -C build/ ./

**Задание 1**

Необходимо реализовать подсчет слов в названиях статей Википедии (файл с названиями статей: **titles-a**). Для начала работы предоставляется шаблон TitleCount.java, в котором необходимо реализовать исходный код в местах помеченных //TODO (код помеченный комментарием «Don't change», трогать не надо).

Вашему приложению из входного списка с названиями статей Википедии (в каждой строке по одному названию) необходимо выделить отдельные слова, используя список разделителей (файл **delimiters.txt**), перевести все слова в нижний регистр, удалить общие слова из списка (файл **stopwords.txt**) и подсчитать количество оставшихся слов, сохранив результат (порядок слов неважен). Скомпилированное и упакованное приложение можно будет запустить используя команду следующего вида (пути HDFS, естественно, могут отличаться):

# hadoop jar TitleCount.jar TitleCount -D

stopwords=/lab1/1/stopwords.txt -D delimiters=/lab1/1/delimiters.txt

/lab1/1/titles-a /lab1/1-output

**Задание 2**

Необходимо реализовать нахождение топ N (N задается в качестве параметра) слов из названий статей Википедии (файл с названиями статей: **titles-b**). Для начала работы предоставляется шаблон TopTitles.java, в котором необходимо реализовать исходный код в местах помеченных //TODO (код помеченный комментарием «Don't change», трогать не надо).

Вашему приложению из входного списка с названиями статей Википедии (в каждой строке по одному названию) необходимо выделить отдельные слова, используя список разделителей (файл **delimiters.txt**), перевести все слова в нижний регистр, удалить общие слова из списка (файл **stopwords.txt**), подсчитать количество оставшихся слов и выбрать из них топ N слов, сохранив результат.

Следует использовать следующий метод сортировки:

1. Отсортировать список пар «ключ-значение» по возрастанию, сначала по значению (количество слов), а при совпадении — по ключу (сами слова). Строки сортируются лексикографически.
2. Последние N пар из полученного списка и будут необходимыми топ N словами.

Например, для списка {"A": 100, "B": 99, "C": 98,"D": 97,"E": 96, "F": 96, "G": 90},отсортированные вариант будет:

"G": 90

**"E": 96**

**"F": 96**

"D": 97

"C": 98

"B": 99

"A": 100

а топ N слов: A, B, C, D, F.

**Примечания**: 1) для промежуточных данных MapReduce заданий предлагается готовый класс TextArrayWriter; 2) для облегчения сортировки предлагается готовый класс Pair (например, можно использовать его в паре с Java-коллекцией SortedSet; 3) полезную работу по отсеиванию ненужных слов (не входящих в топ) нужно частично сделать **в методе cleanup** класса TopTitlesMap, а затем **в методе reduce** класса TopTitlesReduce (тем самым значительно уменьшится количество пересылаемых данных между фазами Map и Reduce).

Скомпилированное и упакованное приложение можно будет запустить используя команду следующего вида (пути HDFS и N, естественно, могут отличаться):

# hadoop jar TopTitles.jar TopTitles -D

stopwords=/lab1/2/stopwords.txt -D delimiters=/lab1/2/delimiters.txt

-D N=5 /lab1/2/titles-b /lab1/2-output

**Задание 3**

Необходимо реализовать нахождение статистики топ слов из названий статей Википедии (файл с названиями статей: **titles-c**). Для начала работы предоставляется шаблон TopTitleStatistics.java, в котором необходимо реализовать исходный код в местах помеченных //TODO (код помеченный комментарием «Don't change», трогать не надо).

Вашему приложению из входного списка с названиями статей Википедии (в каждой строке по одному названию) необходимо выделить отдельные слова, используя список разделителей (файл **delimiters.txt**), перевести все слова в нижний регистр, удалить общие слова из списка (файл **stopwords.txt**), подсчитать количество оставшихся слов и выбрать из них топ N слов.

Следует использовать метод сортировки, описанный в предыдущем разделе.

Далее необходимо сохранить в качестве результата следующую статистику по топ N словам: «Mean» - среднее, «Sum» - сумму всех топ N слов, «Max», «Min» - максимум и минимум, «Variance» - дисперсию. Результат нужно округлять до целого (самое простое — использовать в вычислениях встроенный тип Integer, т. е. неявное преобразование).

Скомпилированное и упакованное приложение можно будет запустить используя команду следующего вида (пути HDFS и N, естественно, могут отличаться):

# hadoop jar TopTitleStatistics.jar TopTitleStatistics -D

stopwords=/lab1/3/stopwords.txt -D delimiters=/lab1/3/delimiters.txt

-D N=5 /lab1/3/titles-c /lab1/3-output

**Содержание отчета**

1. Цель работы.
2. Исходный код и результат (вывод 10-20 слов) задания 1.
3. Исходный код и результат (для N=10) задания 2.
4. Исходный код и результат (для N=50) задания 3.
5. Выводы.

**Общие требования к оформлению**

В отчете приводится описание всех выполненных действий согласно требованиям к содержанию.

Отчет сдается в электронном виде (высылается на электронную почту как минимум за неделю до очной сдачи).

Обычный текст - шрифт 14 пт., заголовок — 16 пт., междустрочный интервал — одинарный, красная строка — 1 см, поля по 2 см.

Пример титульного листа можно найти на сервере кафедры или взять из выполненной работы по другому курсу. Обязательно должна быть информация: «Лабораторная/Курсовая работа по курсу … на тему … Выполнили … Проверил: ...».

В конце работы обязателен раздел «Список использованной литературы» или «Библиографический список», в случае использования сторонних источников.