Урок 2. Концепция MapReduce. Управление ресурсами через YARN

Выполнил Колеганов Н.Д.

Задание

0. [Исследовательское задание] Как получить доступ под пользователем hdfs в файловую систему не имея sudo?
1. Опробовать запуски map-reduce задач для кластера используя hadoop-mapreduce-examples.jar.
1. Выполнить три любых задачи включенных в этот JAR.
2. Найти свои задачи в интерфейсе Cloudera Manager
3. Опробовать навигацию по интерфейсу YARN
4. Сделать документ со скриншотами того, чтовы видели.
2. [Факультативное, для тех кто знает JAVA] Собрать программу для MR на Java и запустить ее. Wordcount будет вполне достаточен.
3. [Задание на 5++] Повторить вот этот пример https://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/

Это задание автоматически закрывает все предыдущие. Если удастся, то пункты 1 и 2 делать не нужно.

Решение

0. export HADOOP_USER_NAME=<your hdfs user>

1)Расчет числа пи

```
]$ export YARN_EXAMPLES=/opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/hadoop-mapreduce
]$ yarn jar $YARN_EXAMPLES/hadoop-mapreduce-examples.jar  pi 32 20000
```

```
Number of Maps  = 32
Samples per Map = 20000
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Wrote input for Map #10
Wrote input for Map #11
Wrote input for Map #12
Wrote input for Map #13
Wrote input for Map #14
Wrote input for Map #15
Wrote input for Map #16
Wrote input for Map #17
Wrote input for Map #18
Wrote input for Map #19
Wrote input for Map #20
Wrote input for Map #21
Wrote input for Map #22
Wrote input for Map #23
Wrote input for Map #24
Wrote input for Map #25
Wrote input for Map #26
Wrote input for Map #27
Wrote input for Map #28
Wrote input for Map #29
Wrote input for Map #30
Wrote input for Map #31
```

```
09:58:53 INFO client.RMProxy: Connecting to ResourceManager at manager.novalocal/89.208.221.132:8032
09:58:54 INFO input.FileInputFormat: Total input paths to process : 32
09:58:54 INFO mapreduce.JobSubmitter: number of splits:32
09:58:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1579555543948_0040
09:58:55 INFO impl.YarnClientImpl: Submitted application application_1579555543948_0040
09:58:56 INFO mapreduce.Job: The url to track the job: http://manager.novalocal:8088/proxy/application_1579555543948_0040/
09:58:56 INFO mapreduce.Job: Running job: job_1579555543948_0040
09:59:03 INFO mapreduce.Job: Job job_1579555543948_0040 running in uber mode : false
```

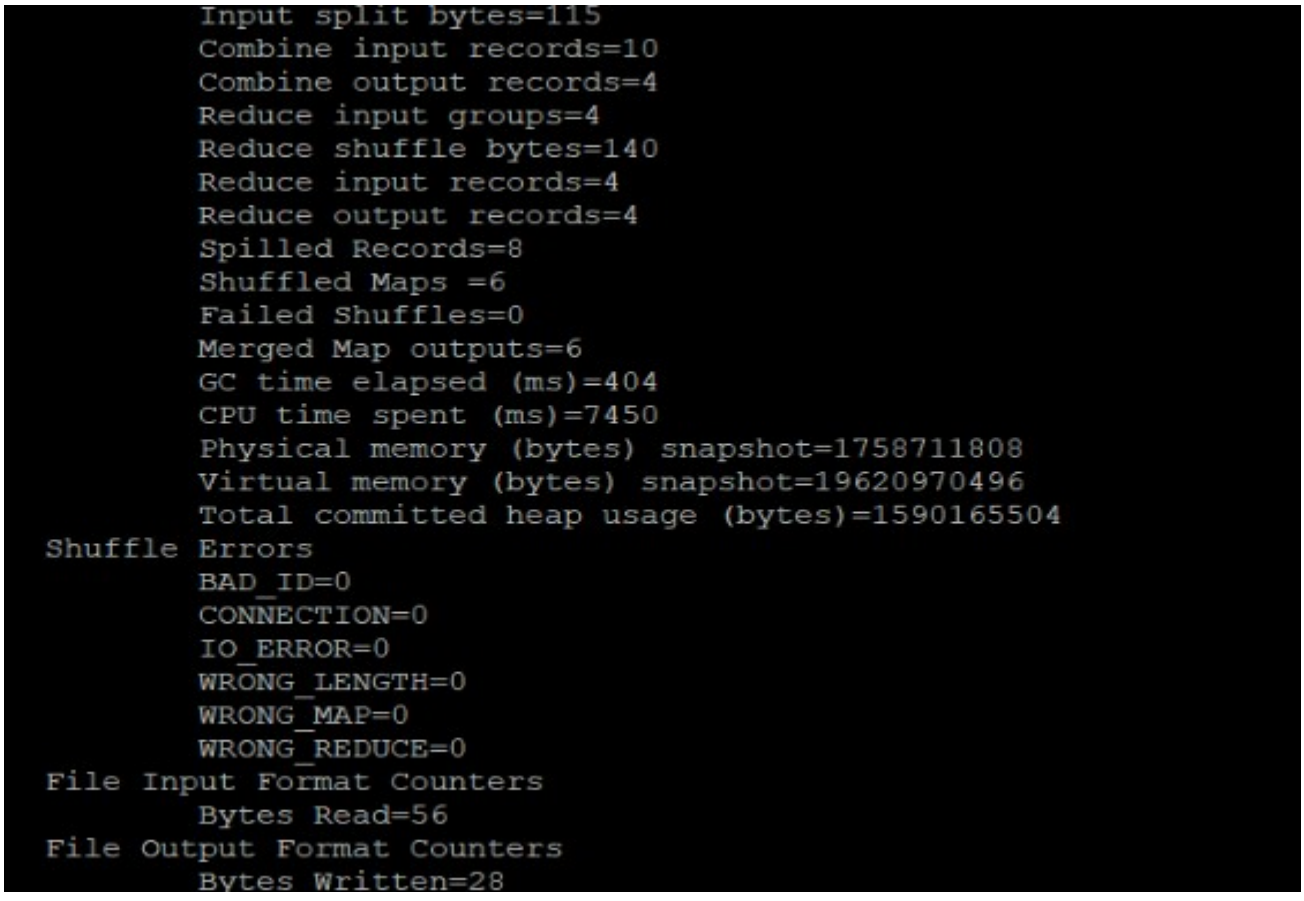
```
                Total time spent by all maps in occupied slots (ms)=90376
                Total time spent by all reduces in occupied slots (ms)=8885
                Total time spent by all map tasks (ms)=90376
                Total time spent by all reduce tasks (ms)=8885
                Total vcore-milliseconds taken by all map tasks=90376
                Total vcore-milliseconds taken by all reduce tasks=8885
                Total megabyte-milliseconds taken by all map tasks=92545024
                Total megabyte-milliseconds taken by all reduce tasks=9098240
        Map-Reduce Framework
                Map input records=32
                Map output records=64
                Map output bytes=576
                Map output materialized bytes=1120
                Input split bytes=5174
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=1120
                Reduce input records=64
                Reduce output records=0
                Spilled Records=128
                Shuffled Maps =32
                Failed Shuffles=0
                Merged Map outputs=32
                GC time elapsed (ms)=2115
                CPU time spent (ms)=24760
                Physical memory (bytes) snapshot=15027654656
                Virtual memory (bytes) snapshot=92171190272
                Total committed heap usage (bytes)=14887157760
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=3776
        File Output Format Counters
                Bytes Written=97
Job Finished in 81.842 seconds
Estimated value of Pi is 3.14153125000000000000
```

2) Подсчет слов

```
$ yarn jar $YARN_EXAMPLES/hadoop-mapreduce-examples.jar  wordcount /t_f_2/words_test.txt /t_f_2/output
```

```
10:20:30 INFO client.RMProxy: Connecting to ResourceManager at manager.novalocal/89.208.221.132:8032
10:20:31 INFO input.FileInputFormat: Total input paths to process : 1
10:20:31 INFO mapreduce.JobSubmitter: number of splits:1
10:20:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1579555543948_0042
10:20:31 INFO impl.YarnClientImpl: Submitted application application_1579555543948_0042
10:20:31 INFO mapreduce.Job: The url to track the job: http://manager.novalocal:8088/proxy/application_1579555543948_0042/
10:20:31 INFO mapreduce.Job: Running job: job_1579555543948_0042
10:20:37 INFO mapreduce.Job: Job job_1579555543948_0042 running in uber mode : false
10:20:37 INFO mapreduce.Job:  map 0% reduce 0%
10:20:43 INFO mapreduce.Job:  map 100% reduce 0%
10:20:48 INFO mapreduce.Job:  map 100% reduce 33%
10:20:51 INFO mapreduce.Job:  map 100% reduce 50%
10:20:52 INFO mapreduce.Job:  map 100% reduce 67%
10:20:55 INFO mapreduce.Job:  map 100% reduce 83%
10:20:56 INFO mapreduce.Job:  map 100% reduce 100%
10:20:57 INFO mapreduce.Job: Job job_1579555543948_0042 completed successfully
10:20:57 INFO mapreduce.Job: Counters: 49
```

```
File System Counters
        FILE: Number of bytes read=164
        FILE: Number of bytes written=1046426
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=171
        HDFS: Number of bytes written=28
        HDFS: Number of read operations=21
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=12
Job Counters
        Launched map tasks=1
        Launched reduce tasks=6
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=3652
        Total time spent by all reduces in occupied slots (ms)=16440
        Total time spent by all map tasks (ms)=3652
        Total time spent by all reduce tasks (ms)=16440
        Total vcore-milliseconds taken by all map tasks=3652
        Total vcore-milliseconds taken by all reduce tasks=16440
        Total megabyte-milliseconds taken by all map tasks=3739648
        Total megabyte-milliseconds taken by all reduce tasks=16834560
Map-Reduce Framework
        Map input records=4
```



3) Судоку

```
8 5 ? 3 9 ? ? ? ?
? ? 2 ? ? ? ? ? ?
? ? 6 ? 1 ? ? ? 2
? ? 4 ? ? 3 ? 5 9
? ? 8 9 ? 1 4 ? ?
3 2 ? 4 ? ? 8 ? ?
9 ? ? ? 8 ? 5 ? ?
? ? ? ? ? ? 2 ? ?
? ? ? ? 4 5 ? 7 8
```

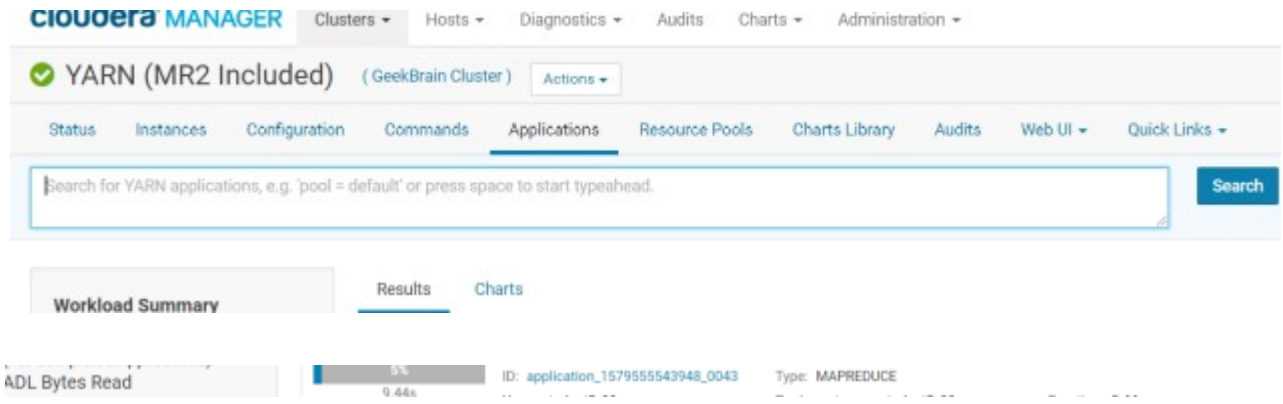
```
$ yarn jar $YARN_EXAMPLES/hadoop-mapreduce-examples.jar sudoku sudoku.dta
Solving sudoku.dta
8 5 1 3 9 2 6 4 7
4 3 2 6 7 8 1 9 5
7 9 6 5 1 4 3 8 2
6 1 4 8 2 3 7 5 9
5 7 8 9 6 1 4 2 3
3 2 9 4 5 7 8 1 6
9 4 7 2 8 6 5 3 1
1 8 5 7 3 9 2 6 4
2 6 3 1 4 5 9 7 8

Found 1 solutions
```

Очень странно но судоку удалось решить только указав файл из файловой системы сервера а не из hdfs. Wordcount съел файл из hdfs a sudoku не видит файл из hdfs.

05/31/2020 1:59 PM

2. Посмотрел окошко с задачами. В этот момент запущен расчет числа пи. Номера задач не совпадают потому что в первый раз не успел зайти в cloudera пока выполнялось.



3. Сделал mapper и reducer

```
[student4_2@manager ~]$ cat /home/student4_2/mapper.py
```

```
#!/usr/bin/env python
"""mapper.py"""

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        print '%s\t%s' % (word, 1)
```

```
[student4_2@manager ~]$ cat /home/student4_2/reducer.py
```

```
#!/usr/bin/env python
"""reducer.py"""

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    print '%s\t%s' % (current_word, current_count)
```

```
student4_2@manager ~]$ echo "foo foo" | /home/student4_2/mapper.py
```

```
oo      1
oo      1
```

```
student4_2@manager ~]$ echo "foo foo qwe qwe foo" | /home/student4_2/mapper.py | sort
```

```
oo      1
oo      1
oo      1
we      1
we      1
```