Database Technologies Project – Phase 3

Overview

In the third phase of the project, we explore the use of distributed database systems and technologies to coordinate queries and operations on multiple nodes. The objectives of this phase are -

1. Write a Map/Reduce program
2. Perform a distributed query and a distributed transation between two heterogenous database systems

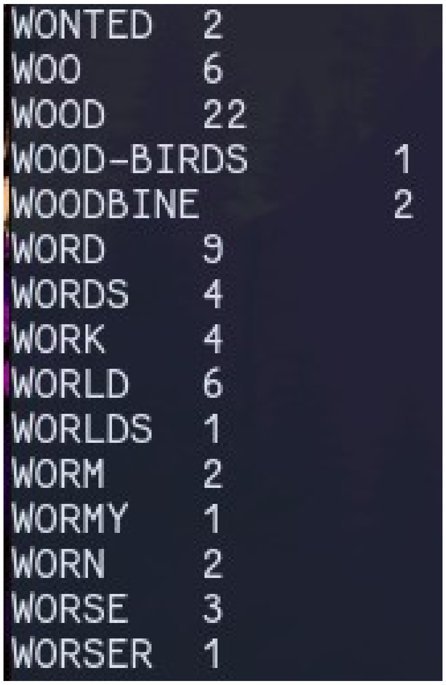
The entire project can be found at https://github.com/anihm136/DBMS_Project. Relevant SQL files are uploaded in the project submission folder as well. A Hadoop instance in distributed mode, running on a cluster of Docker containers has been used for running the Map/Reduce program, and SQL Server 2017 in Windows has been used for the distributed query.

A. Map/Reduce program

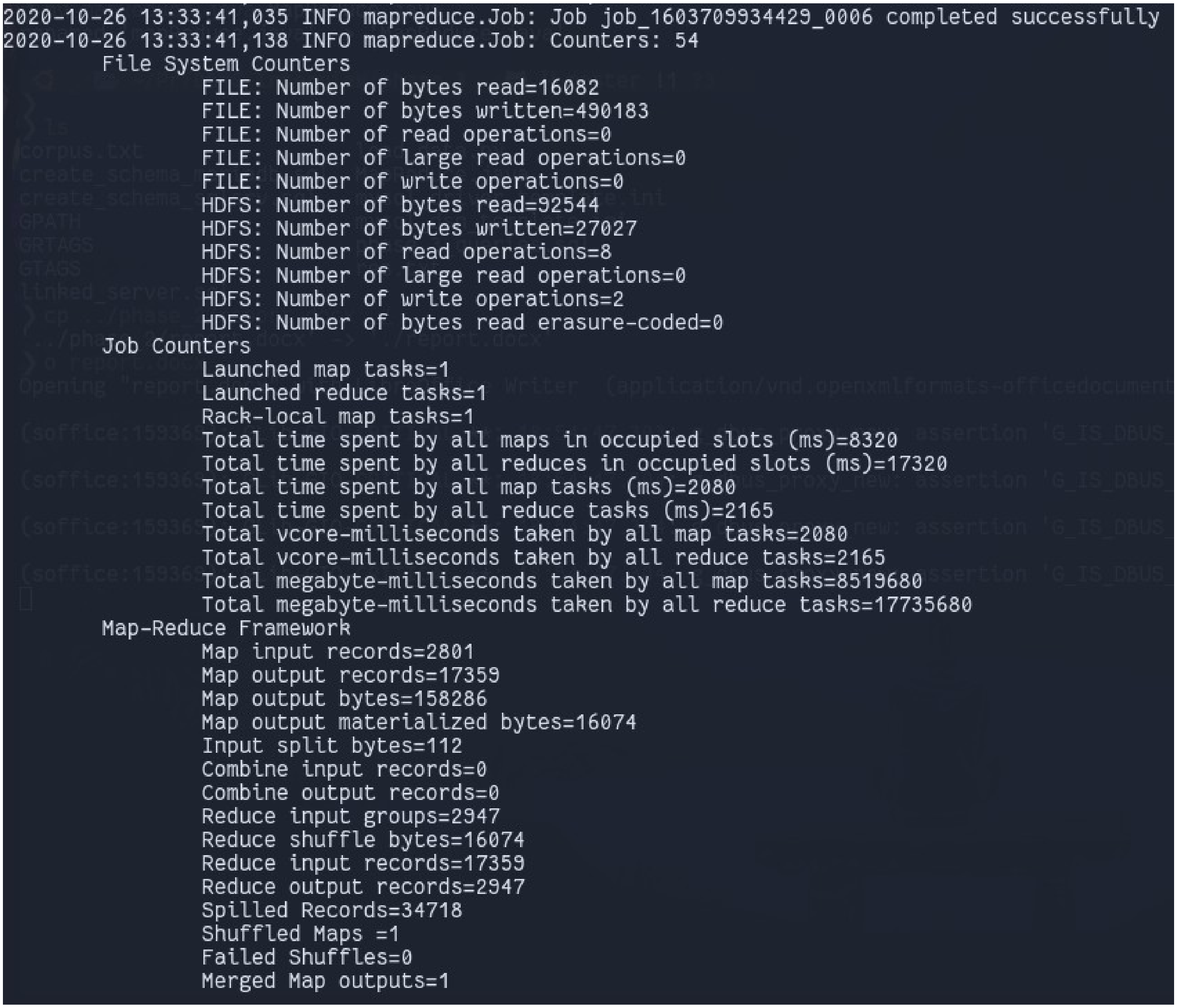
A simple Map/Reduce program for Apache Hadoop, written in Java. It counts the number of occurrences of various words in a text corpus. The two phases are -

1. Map: Tokenize the corpus, clean up each word to prevent duplicates and map each word to a single number (1) to count it
2. Reduce: Group all entries with the same key (I.e, same word) and add the values (I.e, 1). The result will be the number of occurrences of each word

The source code as well as the result has been uploaded to the submisison directory. The entire text of *A Midsummer Night's Dream* by William Shakespeare has been used as a corpus. Some images of the execution logs are as follows -



```
WONTED  2
WOO     6
WOOD    22
WOOD-BIRDS      1
WOODBINE        2
WORD    9
WORDS   4
WORK    4
WORLD   6
WORLDS  1
WORM    2
WORMY   1
WORN    2
WORSE   3
WORSER  1
```



```
2020-10-26 13:33:41,035 INFO mapreduce.Job: Job job_1603709934429_0006 completed successfully
2020-10-26 13:33:41,138 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=16082
                FILE: Number of bytes written=490183
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=92544
                HDFS: Number of bytes written=27027
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=8320
                Total time spent by all reduces in occupied slots (ms)=17320
                Total time spent by all map tasks (ms)=2080
                Total time spent by all reduce tasks (ms)=2165
                Total vcore-milliseconds taken by all map tasks=2080
                Total vcore-milliseconds taken by all reduce tasks=2165
                Total megabyte-milliseconds taken by all map tasks=8519680
                Total megabyte-milliseconds taken by all reduce tasks=17735680
        Map-Reduce Framework
                Map input records=2801
                Map output records=17359
                Map output bytes=158286
                Map output materialized bytes=16074
                Input split bytes=112
                Combine input records=0
                Combine output records=0
                Reduce input groups=2947
                Reduce shuffle bytes=16074
                Reduce input records=17359
                Reduce output records=2947
                Spilled Records=34718
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
```

B. Distributed query

Distributed queries are supported by SQL Server using linked databases. An external database can be linked to an SQL Server instance using one of many data sources provided by Microsoft. These external databases can then be queried and the results can be viewed and used in SQL Server. Further, the external databases can also participate in transactions originating in SQL Server, managed by SQL Server.
In this particular example, a MariaDB database is linked to an SQL Server database using an ODBC DSN as the data source. A distributed transaction is run from the SQL Server database, including a query on the MariaDB database.

The MariaDB server contains the CANDIDATE table, containing information about interview candidates. The SQL Server instance contains the JOB_DETAILS and JOB_ROLE tables, which contain information about the jobs they are applying for. A distributed query is set up to find candidates per location, requiring a join between the candidates table in MariaDB and the two job tables in SQL Server. All relevant SQL files are attached, including the schema for each database, the setup of the linked server and the query