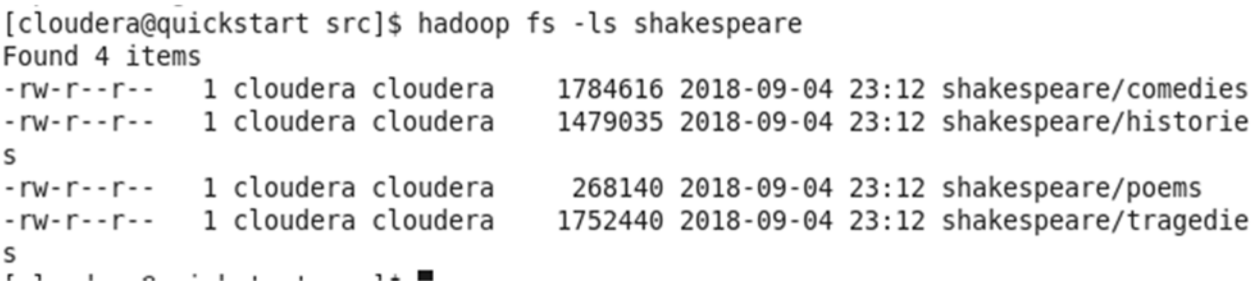
Adrien Xie 451852 xie.z@wustl.edu

Problem 1

(a) List all the files in the Shakespeare folder in HDFS

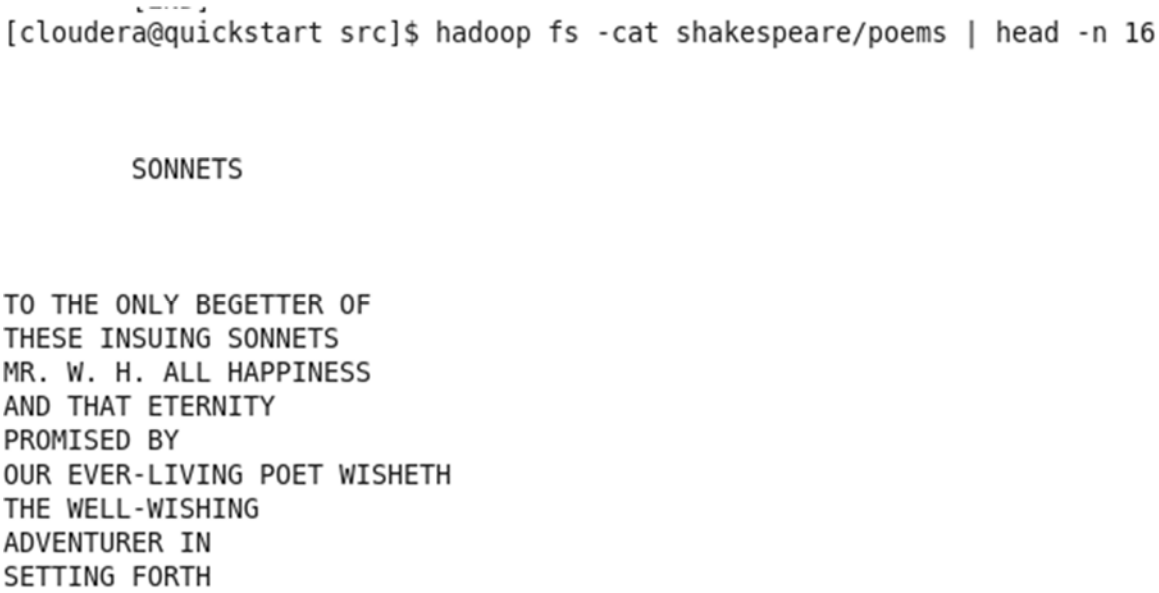
$hadoop fs -ls Shakespeare



```
[cloudera@quickstart src]$ hadoop fs -ls shakespeare
Found 4 items
-rw-r--r--   1 cloudera cloudera    1784616 2018-09-04 23:12 shakespeare/comedies
-rw-r--r--   1 cloudera cloudera    1479035 2018-09-04 23:12 shakespeare/historie
s
-rw-r--r--   1 cloudera cloudera     268140 2018-09-04 23:12 shakespeare/poems
-rw-r--r--   1 cloudera cloudera    1752440 2018-09-04 23:12 shakespeare/tragedie
s
```

Display the first 16 lines of the poems data in HDFS

$hadoop fs -cat Shakespeare/poems | head -n 16



```
[cloudera@quickstart src]$ hadoop fs -cat shakespeare/poems | head -n 16



        SONNETS



TO THE ONLY BEGETTER OF
THESE INSUING SONNETS
MR. W. H. ALL HAPPINESS
AND THAT ETERNITY
PROMISED BY
OUR EVER-LIVING POET WISHETH
THE WELL-WISHING
ADVENTURER IN
SETTING FORTH
```

(b) Large replication factor means more replicated files. These duplicated will occupy more memory space on the data node, which is of limited capacity. Therefore one downside of large replication factor is that it will exploit the data nodes' memory capacities.
Additionally, too much of files will lead to the generation of too much meta-data and storing these meta-data in the name node will become a challenge since name node has limited memory capacity as well.

(c) One benefit of large block size is that this will result in less meta-data files and hence requires smaller capacity name node. Otherwise, name node will be overflowed with meta-data of huge amounts of small blocks.
One disadvantage is that for small number of tasks, large block size will result in too much time spent on transferring and hence the job will run slower than they could.

(d) Because $128 \times 5 < 642 < 128 \times 6$, we will have the file stored in six blocks

RA	RB	RC
N1 Part 1, Part 4	N3 Part 3, Part 1	N5 Part 5, Part 2
N2 Part 2, Part 5	N4 Part 4, Part 6	N6 Part 6, Part 3

Meta-Data

File-Part 1: RA-N1, RB-N3

File-Part 2: RA-N2, RC-N5

File-Part 3: RB-N3, RC-N6

File-Part 4: RB-N4, R1-N1

File-Part 5: RC-N5, RA-N2

File-Part 6: RC-N6, RB-N4

(e) Two disadvantages of HDFS

1. Low-latency data access

   HDFS is designed to deliver a high throughput of data and this may be at the expense of latency. Therefore applications that require low-latency access to data will not work well with HDFS.

2. Because the name node holds file system metadata in memory, the limit to the number of files in a file system is governed by the amount of memory on the name node. Therefore HDFS cannot store too many small files because they will produce way too much metadata that will overflows name node' memory.

(f) Download from HDFS to local file. For example, download the Shakespeare file to a new local directory downloadedfile

```
$hadoop fs –get Shakespeare downloadedfile
```

The transferring process begin with the computer communicating with the name node to look up the metadata for the file. Once the name node send back metadata, the computer can directly access the data nodes and begin the data transferring process.

Problem 2

Mapper Input	Mapper Output	Reducer Input	Reducer Output
15	(3, 15), (5,15)	2, [24, 30]	2, 54
21	(3, 21), (7,21)	3, [15, 21, 24, 30]	3, 90
24	(2, 24), (3, 24)	5, [15, 30]	5, 45
30	(2, 30), (3, 30), (5, 30)	7, [21, 49]	7, 70
49	(7, 49)		

Problem 3

(a)

Mapper output	Reducer input	Reducer output
gif, 1200	gif, [1200, 1900]	gif, 1550
html, 900	html, [900, 1100]	html, 1000
gif, 1900	jpg, [4000]	jpg, 4000
jpg, 4000		
html, 1100		

(b) The keys are represented by string and the values are presented by integers.