

Hadoop Assignment

Question 1 (40 %)

submit the source code as well as the statistics/results you gathered when running the Hadoop job on shakespeare corpus.

Download the following three works by Shakespeare from Project Gutenberg into a local directory (e.g. shakespeare1):

- <http://www.gutenberg.org/cache/epub/1524/pg1524.txt>
- <http://www.gutenberg.org/cache/epub/1112/pg1112.txt>
- <http://www.gutenberg.org/cache/epub/2267/pg2267.txt>

and upload the directory to your HDFS home directory.

1. With reference to WordCount.java. Modify the program again to output the following statistics over the Shakespeare corpus:

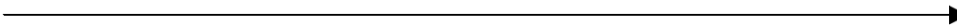
- the number of unique (or distinct) terms in the corpus
- the number of words in the corpus that start with the letter T/t
- the number of terms appearing less than 5 times
- the number of files read overall
- the 5 most often occurring terms and their frequency in the corpus.

The output of the job in the end should be the five terms that occur most frequently in the corpus together with their actual frequency. Submit the source code and the job output. Report the numbers you got for each of the queries listed above.

2. Starting off with the original WordCount.java again, write a Hadoop job that computes the number of terms appearing in only one document of the corpus. A term may appear multiple times in a single document. Only if the term does not appear in any other document of the corpus, should it be included in the count. Submit your source code and report how many terms appear only in one document of our shakespeare corpus.

Question 2 (30 %) Transactions

Consider the following schedule involving three transactions T1, T2 and T3:

	1	2	3	4	5	6	7	8	9	10
										
T1:	W(A)	R(B)	R(C)		W(B)					
T2:							R(C)	W(B)	W(C)	
T3:				W(B)		W(C)				R(A)

- (1) Draw the precedence graph for this schedule.
- (2) Is this schedule conflict serializable? Why or why not? If it is conflict serializable, give the equivalent serial schedule (just write the order of the transactions).
- (3) Write down all instances where one transaction “reads from” another transaction.
(If T2 reads from T1, write T1 → T2.)
Now swap the 4th and 5th actions in the schedule.
- (4) Is the new schedule conflict serializable? Why or why not? If it is conflict serializable, give the equivalent serial schedule (just write the order of the transactions).

Question 3 (30 %) Recovery

- (a) What is done during Analysis?
- (b) What is done during Redo?
- (c) What is done during Undo?

LSN	LOG
00	Begin Check Point
10	End Check Point
20	Update: T1 writes P5
30	Update: T2 writes P3
40	T2 Commit
50	T2 End
60	Update: T3 writes P3
70	T1 Abort
80	Update: T4 writes P6
90	Update: T4 writes P7
100	T4 Commit
110	T4 End
120	Crash