MapReduce: IDF Performance Analysis

1. Data Processing Estimation-

In our implementation, the mapper passes the output directly to the combiner which processes the information while the mapper processes continue running. Thus, the actual time the reducer runs overlaps with the time that the mapper runs. So, the best estimate comes from scaling up total job time. For a 104 MB file, the total time taken was 44.4174 s
104/44.4174 = 2.34 MB/s. Therefore in 10 minutes, 2.34 x 600 = 1404.86 MB

2. Profiling results-

- 1MB
  - Initialization: 0.0156s
  - Mapper Time: 0.4819s
  - Reducer/Shuffle Time: 0.5063s
  - Total Time: 0.5261s
- 10MB
  - Initialization: 0.0143s
  - Mapper Time: 4.6288s
  - Reducer/Shuffle Time: 4.6552s
  - Total Time: 4.6734s
- 100MB
  - Initialization: 0.0144s
  - Mapper Time: 44.3898s
  - Reducer/Shuffle Time: 44.4174s
  - Total Time: 44.4360s

3. FLOPs Analysis-

- Estimated Computational Complexity: In the implementation, mapper simply does string manipulation, while reducer and combiner do integer addition. Therefore only final output has floating point operations.
  - idf = log(num_docs.value / doc_freq[key]), so 1 division and 1 log.
  - Therefore, assuming vocabulary of size V, 2*V FLOPS
- Actual Computational Complexity:
  - In 100 MB file, V = 150, t = 44.436s
  - Actual FLOPS = Expected FLOPS/time taken => 2*150/44.436 = 6.75