Homework 10

Please hand in as a pdf. Due Dec 5, 2024

Problem 1:

The National Climatic Data Center (NCDC) maintains datasets for a variety of data. The maritime dataset https://www.ncei.noaa.gov/data/global-marine/archive/ has information on historical buoy measurements.

(a) Download and decode a sample file. In particular, use longitude and latitude to locate some measurement locations on a world map.
(b) The data describe wind speed and direction. Lay out a map-reduce algorithm to calculate prevailing wind speeds globally depending on the season. In particular, design mappers, combiners, and reducers. (No code required).

Problem 2:

HDFS has a high-availability mode where a namenode has a standby backup. It has a Quorum Journal Manager (QJM) for providing a highly available edit log, consisting of several servers and enforcing that an entry is written if and only if it is written to a majority of these servers.

Are there ways to achieve consistency that can exploit the properties of a log?

Problem 3:

Assume that we want to provide totally ordered multicast using unreliable uni-directional messages. (Messages can be delayed, can be lost, can be duplicated). This is part of a communication middleware layer on top of TCP/UDP. A message that is being handed up to the application layer is lost to the communication middleware layer.

(a) One design uses a token. Only the process with the token is allowed to send. After a process is done sending, it hands the token to the next server. Show how you can guarantee that all messages arrive at the various application processes in order.

(b) Processes can fail. How can you deal with the failure of a process that is not in possession of the token?

(c) How can the other processes deal with a failure of the process with the token.

Problem 4:

Assume that you have a small relational database table $R(A, B, C)$, maybe 1000 entries, with three numerical attributes. Assume that you have a very large, distributed relational table $S(A, D, E, F)$ of size petabytes divided over many sites. How would you use map-reduce in order to calculate $R \bowtie_{R.A<S.A} (B, D)$, that is, all values (b, d) such that there exists values

$(a, b, c) \in R$, values $(a', d, e, f) \in S$ with $a < a'$? Minimize data movement and assume that the result is small.