DevOps / Docker Challenge

Map reduce is a common technique to process large data sets. A common basic example is counting the number of words in a large text.

Problem: You have a *large* text file containing text lines in any real world language. Lines are considered to be average in length so edge cases such as a file with just two very large lines should still work. Remember to use a real language and not just made up character streams. Count the number of occurrences of each word in the text and produce the results as output.

Use docker-machine from Docker to spin up hosts running a number of containers that process the input file and process it in a distributed way using a map-reduce algorithm. The input text must not be in the context of the containers and must be input into the system through a socket connection. The final output is a list of words and the number of occurrences for each, also available outside the context of the docker system through a socket. You do not need to run the different containers in actual physical machines but you must run them in different Docker hosts by leveraging docker-machine.

Use any common language (for instance C, C++, Java or Go...) for your main functionality code though remember some other language can be used for integration glue. Please use default language libraries only, no batch, stream processing or systems provisioning frameworks. Be as efficient as possible while avoiding using high-level library routines (remember, *do not use* any github, Hadoop, Spark, Flink, Storm, or any other externally provided solutions). Provide a rationale for your approach. Design schemas are welcome.

Expectations: make sure you obtain a large text file and process it before handing over your assignment code. Think about how to best verify your results and how to practically get input and output of the containers.

Bonus: sort the output by occurrence so the most common words come first