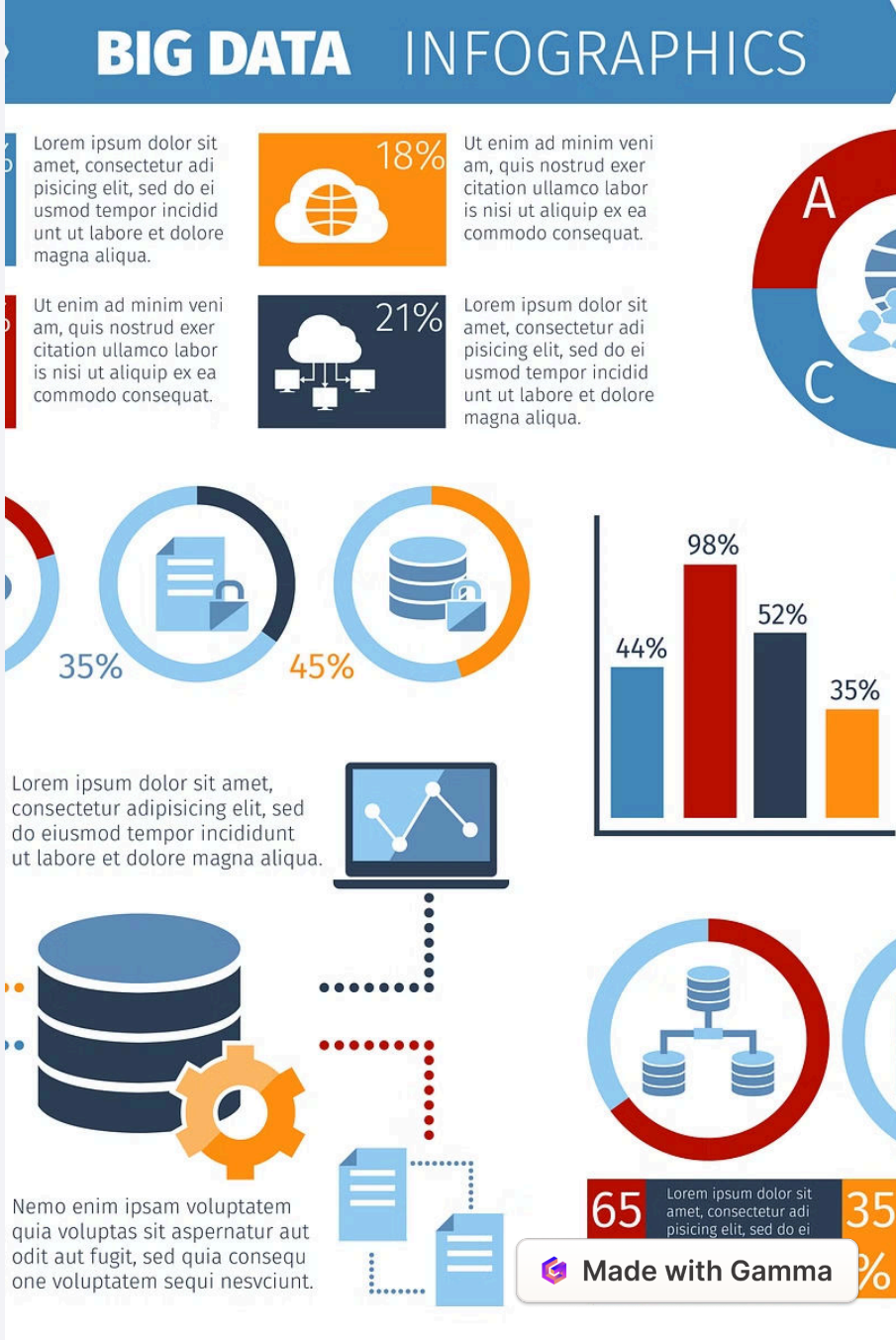
Word Count and Most Frequent Words

The process of counting the frequency of words using MapReduce and extracting the most frequent ones is fundamental in big data processing. This guide provides detailed instructions and insights for students or professionals learning about MapReduce and its application in handling large volumes of data.

SA **by Sarika Alladi**



Word Count: Mapper Class

Functionality

The Mapper class tokenizes input lines, generating individual words, and creates key-value pairs with a count of 1 for each word.

Implementation

It involves overriding the map() method to process input data and emit the intermediate key-value pairs.

4. If the reducer position differs from the positions shown below, oil volumes may vary. Consult Vortex engineering.
5. If equipped with a backstop, mounting positions C and D will need more oil than listed above. Backstops do not have rolling elements but must slide on the shaft, therefore the input shaft must stay lubricated. Increase the oil within the reducer until it submerges half of the input shaft. This increased oil quantity may reduce the thermal capacity of the reducer. It is recommended that you monitor the reducer for signs of overheating.

Standard Mounting Positions

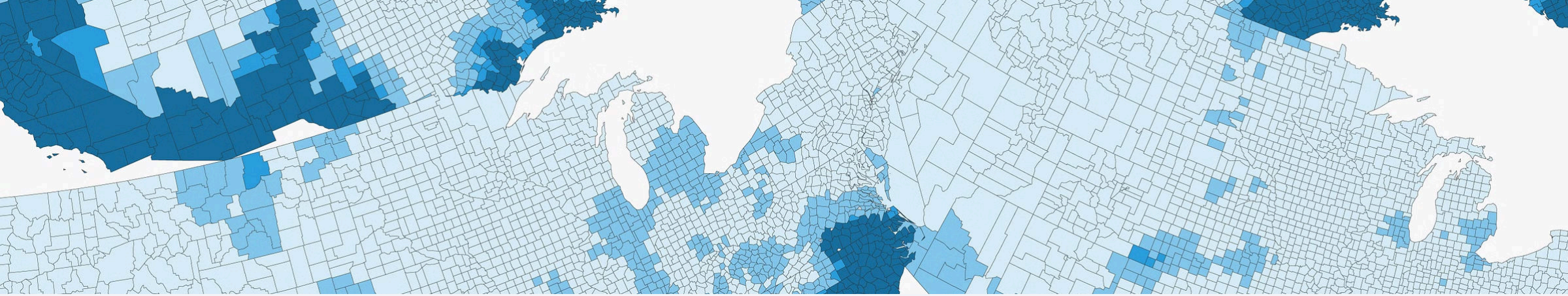
Word Count: Reducer Class

1 Aggregation

The Reducer class aggregates the word counts received from the Mapper, combining the counts for each word.

2 Execution

It entails overriding the reduce() method and processing the intermediate key-value pairs to generate the final word counts.



Most Frequent Words: Mapper Class

1

Function

The Mapper reads the Word Count job output, extracts words with their counts, and emits them as key-value pairs with the count as the key.

Most Frequent Words: Reducer Class

Sorting

The Reducer sorts the key-value pairs based on counts and emits the most frequent words with their respective counts.

Implementation

It involves implementing the reduce() method to process the key-value pairs from the Mapper and extract the most frequent words.

Chaining MapReduce Jobs

1

Modification

The Driver class is modified to chain the Word Count and Most Frequent Words Count jobs by using the Word Count job output as input for the next job.

Handling Input and Output Paths

1 Driver Class

The new Driver class ensures proper handling of input and output paths for both the Word Count and Most Frequent Words Count jobs.

2 Error Handling

Emphasizing on proper error handling to safeguard code execution and ensure data integrity.

JAR File Submission and Code Quality

Compile

Code Compilation

The JAR file is compiled with the classes containing proper error handling, input validation, and detailed documentation.

Submit

Submission

The compiled JAR file is submitted to the Hadoop cluster for execution.