

Assignment B4

Title: Map reduce operation with suitable example using MongoDB

Problem Statement: Write an example of Map reduce using MongoDB

Objective: To understand concept of Map-reduce as data processing paradigm for condensing large volumes of data into useful aggregated results

Learning Outcome:

Students will be able to understand and effectively implement Map-reduce to obtain useful aggregated results in MongoDB

Hardware and software requirements:

64 bit Linux based OS, MongoDB 2.6+

Concept related theory:

Map-Reduce

- Map-Reduce is a powerful and flexible tool for aggregating data
- It can solve problems that are too complex to express using the aggregation frameworks query language
- It uses JavaScript as its query language
- It is fairly slow and should not be used for real-time data analysis
- Map-Reduce can be easily parallelized across multiple servers

Map Phase

- Map an operation onto every document in a collection. That operation could be either "do nothing" or "emit these keys with X values"

Intermediate stage

- Keys are grouped and lists of emitted values are created for each key

Reduce phase

- Takes this list of values and reduces it to a single element
- This element is returned to the shuffle step until each key has a list containing a single value: the result

Syntax:

```
db.collection.mapReduce(  
  function ( ) { emit (key, value ) },  
  function (key, values ) { return reduceFunction },  
  {  
    out: collection,  
    query: document,  
    sort: document,  
    limit: number  
  })
```

Requirements for the map function:

- Reference the current document as "this" within the function
- It should not access the database for any reason
- It should be pure, i.e. without side effects outside function
- It may optionally call `emit(key, value)` any number of times to create an output document associating key with value

Requirements for the reduce function

- It should not access the database for any reason
- It should not affect the outside system
- The value argument must be an array whose elements are the type value objects that are mapped to the key
- Reduce function can be invoked many times for the same key. In that case, the previous output from the reduce function will become one of the input values to the next reduce function for the same key.
- Reduce function can access the variables defined in the scope parameter
- Inputs to reduce must not be larger than half of MongoDB's maximum BSON document size, i.e. 16MB.

Conclusion:

Thus, we have understood the concept of Map-Reduce and successfully aggregated data using the same