

Engineering Big-Data Systems

Assignment 2:

Name: Abhilash Kosaraju NEUID :001205393 Date: 10/03/2018

2. PROGRAMMING ASSIGNMENT

.bat File for Data Import

```
C:\WINDOWS\system32\cmd.exe
processing C:\Users\soujanya\Desktop\Abhi\NYSE_daily_prices\NYSE_daily_prices_A.csv
Copying data in to the MongoDB collection stocks
2018-10-03T16:08:44.889-0400 connected to: localhost
2018-10-03T16:08:46.882-0400 [#####] stocks.stocks 2.56MB/39.1MB (6.5%)
2018-10-03T16:08:49.880-0400 [#####] stocks.stocks 6.52MB/39.1MB (16.7%)
2018-10-03T16:08:52.880-0400 [#####] stocks.stocks 10.6MB/39.1MB (27.2%)
2018-10-03T16:08:55.884-0400 [#####] stocks.stocks 14.6MB/39.1MB (37.3%)
2018-10-03T16:08:58.880-0400 [#####] stocks.stocks 18.7MB/39.1MB (47.7%)
2018-10-03T16:09:01.880-0400 [#####] stocks.stocks 22.5MB/39.1MB (57.5%)
2018-10-03T16:09:04.879-0400 [#####] stocks.stocks 26.6MB/39.1MB (67.9%)
2018-10-03T16:09:07.879-0400 [#####] stocks.stocks 30.6MB/39.1MB (78.1%)
2018-10-03T16:09:10.880-0400 [#####] stocks.stocks 34.4MB/39.1MB (87.9%)
2018-10-03T16:09:13.880-0400 [#####] stocks.stocks 38.1MB/39.1MB (97.4%)
2018-10-03T16:09:14.652-0400 [#####] stocks.stocks 39.1MB/39.1MB (100.0%)
2018-10-03T16:09:14.652-0400 imported 735026 documents
Press any key to continue . . .
processing C:\Users\soujanya\Desktop\Abhi\NYSE_daily_prices\NYSE_daily_prices_B.csv
Copying data in to the MongoDB collection stocks
2018-10-03T16:09:41.097-0400 connected to: localhost
2018-10-03T16:09:43.089-0400 [#####] stocks.stocks 2.55MB/30.6MB (8.3%)
2018-10-03T16:09:46.089-0400 [#####] stocks.stocks 6.47MB/30.6MB (21.2%)
2018-10-03T16:09:49.089-0400 [#####] stocks.stocks 10.6MB/30.6MB (34.6%)
```

3.1- PROGRAMMING ASSIGNMENT

Average price of stock_price_high values for each stock using MapReduce.

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

{
  "value" : 7.1788237863583
},
{
  "_id" : "ZLC",
  "value" : 28.376730652519726
},
{
  "_id" : "ZHH",
  "value" : 40.11668928696618
},
{
  "_id" : "ZHH",
  "value" : 17.500465430016924
},
{
  "_id" : "ZNT",
  "value" : 25.719687669254412
},
{
  "_id" : "ZQK",
  "value" : 17.905243902438965
},
{
  "_id" : "ZTR",
  "value" : 7.388408084696797
},
{
  "_id" : "ZZ",
  "value" : 9.236293995859228
},
},
"timeMillis" : 92098,
"counts" : {
  "input" : 5454667,
  "emit" : 5454667,
  "reduce" : 3690,
  "output" : 1712
},
"ok" : 1
}

> var map = function(){ emit(this.stock_symbol,this.stock_price_high); }
> var reduce = function(stockKey,values){ var sum =Array.sum(values); var total = values.length; var average = sum/total; return average; }
> db.NYSE_daily_prices.mapReduce(map,reduce,{out :{inline : true}} ).
```

3.2- PROGRAMMING ASSIGNMENT

Using a FINALIZER to find the correct average

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

{
  "_id" : "ZQK",
  "value" : {
    "count" : 5002,
    "cost" : 89562.02999999971,
    "average" : 17.905243902438965
  }
},
{
  "_id" : "ZTR",
  "value" : {
    "count" : 5195,
    "cost" : 38382.77999999986,
    "average" : 7.388408084696797
  }
},
{
  "_id" : "ZZ",
  "value" : {
    "count" : 966,
    "cost" : 8922.260000000015,
    "average" : 9.236293995859228
  }
},
},
"timeMillis" : 149828,
"counts" : {
  "input" : 5454667,
  "emit" : 5454667,
  "reduce" : 3875,
  "output" : 1712
},
"ok" : 1
}

> show collections
NYSE_daily_prices
> var mapFunction1 = function() { emit(this.stock_symbol,{count:1,cost:this.stock_price_high}); }
> var reduceFunction1 = function(keyStock,values){ var value = {count :0, cost:0}; for(i=0;i<values.length;i++){ value.count += values[i].count; value.cost +=values[i].cost; } return value; }
> var finalizeFunction1 = function(keyStock,value){ value.average = (value.cost/value.count); return value; }
> db.NYSE_daily_prices.mapReduce(mapFunction1,reduceFunction1,{ out :{inline :true}, finalize:finalizeFunction1});
```

4- PROGRAMMING ASSIGNMENT

Finalizer to find out the average stock price of each price of all stocks in the finalizer

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
{
  "lcount" : 5195,
  "ocount" : 5195,
  "ccount" : 5195,
  "lcost" : 37786.350000000007,
  "ocost" : 38093.399999999992,
  "ccost" : 38101.799999999984,
  "leverage" : 7.273599615014451,
  "oaverage" : 7.33270452358035,
  "caverage" : 7.334321462945109
},
{
  "_id" : "ZZ",
  "value" : {
    "lcount" : 966,
    "ocount" : 966,
    "ccount" : 966,
    "lcost" : 8595.510000000006,
    "ocost" : 8773.370000000008,
    "ccost" : 8760.920000000002,
    "leverage" : 8.898043478260876,
    "oaverage" : 9.082163561076612,
    "caverage" : 9.069275362318843
  }
},
{
  "timeMillis" : 224123,
  "counts" : {
    "input" : 5454667,
    "emit" : 5454667,
    "reduce" : 4664,
    "output" : 1712
  },
  "ok" : 1
}
> var mapFunction3 = function() { emit(this.stock_symbol,{lcount:1, lcost:this.stock_price_low, ocount:1, ocost: this.stock_price_open, ccount:1, ccost:this.stock_price_close}); }
> var reduceFunction3 = function(keyStock,value) { var value = { lcount : 0, ocount : 0, ccount : 0, lcost : 0,ocost :0, ccost :0 }; for(i=0; i< values.length; i++) { value.lcount += values[i].lcount; value.lcost += values[i].lcost; value.ocost += values[i].ocost; value.ocount += values[i].ocount; value.ccount += values[i].ccost; value.ccost += values[i].ccost; } return value; }
> var finalizeFunction3 = function(keyStock,value){ value.leverage = (value.lcost/value.lcount); value.oaverage = (value.ocost/value.ocount); value.caverage = (value.ccost/value.ccount); return value; }
> db.NYSE_daily_prices.mapReduce(mapFunction3,reduceFunction3,{ out :{inline :true}, finalize:finalizeFunction3});
```

5 & 6 - PROGRAMMING ASSIGNMENT

MongoDB indexing

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db
stocks
> show collections;
stocks
> db.stocks.getIndexes();
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "stocks.stocks"
  }
]
> db.stocks.ensureIndex({"stock_symbol":1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.stocks.ensureIndex({"stock_volume":1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 2,
  "numIndexesAfter" : 3,
  "ok" : 1
}
> db.stocks.getIndexes();
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "stocks.stocks"
  },
  {
    "v" : 2,
    "key" : {
      "stock_symbol" : 1
    },
    "name" : "stock_symbol_",
    "ns" : "stocks.stocks"
  }
]
```

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

```
> db.stocks.getIndexes();
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "stocks.stocks"
  },
  {
    "v" : 2,
    "key" : {
      "stock_symbol" : 1
    },
    "name" : "stock_symbol_1",
    "ns" : "stocks.stocks"
  },
  {
    "v" : 2,
    "key" : {
      "stock_volume" : 1
    },
    "name" : "stock_volume_1",
    "ns" : "stocks.stocks"
  }
]
```

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

```
> db.stocks.getIndexes();
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "stocks.stocks"
  },
  {
    "v" : 2,
    "key" : {
      "stock_symbol" : 1
    },
    "name" : "stock_symbol_1",
    "ns" : "stocks.stocks"
  },
  {
    "v" : 2,
    "key" : {
      "stock_volume" : 1
    },
    "name" : "stock_volume_1",
    "ns" : "stocks.stocks"
  }
]
> db.stocks.find({ "stock_symbol": {"$eq" : "AEA"}}).explain("executionStats");
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "stocks.stocks",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "stock_symbol" : {
        "$eq" : "AEA"
      }
    }
  },
  "winningPlan" : {
    "stage" : "FETCH",
    "inputStage" : {
      "stage" : "IXSCAN",
      "keyPattern" : {
```

```

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
> db.stocks.find({ "stock_symbol": { $eq : "AEA" } }).explain("executionStats");
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "stocks.stocks",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "stock_symbol" : {
        "$eq" : "AEA"
      }
    }
  },
  "winningPlan" : {
    "stage" : "FETCH",
    "inputStage" : {
      "stage" : "IXSCAN",
      "keyPattern" : {
        "stock_symbol" : 1
      },
      "indexName" : "stock_symbol_1",
      "isMultiKey" : false,
      "multiKeyPaths" : {
        "stock_symbol" : [ ]
      },
      "isUnique" : false,
      "isSparse" : false,
      "isPartial" : false,
      "indexVersion" : 2,
      "direction" : "forward",
      "indexBounds" : {
        "stock_symbol" : [
          ["AEA", "AEA"]
        ]
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 1293,
    "executionTimeMillis" : 3,
    "totalKeysExamined" : 1293,
    "totalDocsExamined" : 1293,
  }
}

```

```

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
{
  "executionStats" : {
    "executionSuccess" : true,
    "nReturned" : 1293,
    "executionTimeMillis" : 3,
    "totalKeysExamined" : 1293,
    "totalDocsExamined" : 1293,
    "executionStages" : {
      "stage" : "FETCH",
      "nReturned" : 1293,
      "executionTimeMillisEstimate" : 0,
      "works" : 1294,
      "advanced" : 1293,
      "needTime" : 0,
      "needYield" : 0,
      "saveState" : 10,
      "restoreState" : 10,
      "isEOF" : 1,
      "invalidates" : 0,
      "docsExamined" : 1293,
      "alreadyHasObj" : 0,
      "inputStage" : {
        "stage" : "IXSCAN",
        "nReturned" : 1293,
        "executionTimeMillisEstimate" : 0,
        "works" : 1294,
        "advanced" : 1293,
        "needTime" : 0,
        "needYield" : 0,
        "saveState" : 10,
        "restoreState" : 10,
        "isEOF" : 1,
        "invalidates" : 0,
        "keyPattern" : {
          "stock_symbol" : 1
        },
        "indexName" : "stock_symbol_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
          "stock_symbol" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
      }
    }
  }
}

```

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

```
        "stock_symbol" : [ ]
      },
      "isUnique" : false,
      "isSparse" : false,
      "isPartial" : false,
      "indexVersion" : 2,
      "direction" : "forward",
      "indexBounds" : {
        "stock_symbol" : [
          ["\\AEA\\", "\\AEA\\"]
        ]
      },
      "keysExamined" : 1293,
      "seeks" : 1,
      "dupsTested" : 0,
      "dupsDropped" : 0,
      "seenInvalidated" : 0
    }
  },
  "serverInfo" : {
    "host" : "LAPTOP-DNPE6MFH",
    "port" : 27017,
    "version" : "4.0.2",
    "gitVersion" : "fc1573ba18eee42f97a3bb13b67af7d837826b47"
  },
  "ok" : 1
}
```

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

```
    "version" : "4.0.2",
    "gitVersion" : "fc1573ba18eee42f97a3bb13b67af7d837826b47"
  },
  "ok" : 1
}
> db.stocks.find({ "stock_volume": {$gt : 176400}}).explain("executionStats");
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "stocks.stocks",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "stock_volume" : {
        "$gt" : 176400
      }
    },
    "winningPlan" : {
      "stage" : "FETCH",
      "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {
          "stock_volume" : 1
        },
        "indexName" : "stock_volume_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
          "stock_volume" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
          "stock_volume" : [
            "(176400.0, inf.0]"
          ]
        }
      }
    },
    "rejectedPlans" : [ ]
  },
  "executionStats" : {
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
{
  "docsExamined" : 2586779,
  "alreadyHasObj" : 0,
  "inputStage" : {
    "stage" : "IXSCAN",
    "nReturned" : 2586779,
    "executionTimeMillisEstimate" : 2868,
    "works" : 2586780,
    "advanced" : 2586779,
    "needTime" : 0,
    "needYield" : 0,
    "saveState" : 20825,
    "restoreState" : 20825,
    "isEOF" : 1,
    "invalidates" : 0,
    "keyPattern" : {
      "stock_volume" : 1
    },
    "indexName" : "stock_volume_1",
    "isMultiKey" : false,
    "multiKeyPaths" : {
      "stock_volume" : [ ]
    },
    "isUnique" : false,
    "isSparse" : false,
    "isPartial" : false,
    "indexVersion" : 2,
    "direction" : "forward",
    "indexBounds" : {
      "stock_volume" : [
        "(176400.0, inf.0]"
      ]
    },
    "keysExamined" : 2586779,
    "seeks" : 1,
    "dupsTested" : 0,
    "dupsDropped" : 0,
    "seenInvalidated" : 0
  },
  "serverInfo" : {
    "host" : "LAPTOP-DNPE6MFH",
    "port" : 27017,
  }
}
```

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
{
  "port" : 27017,
  "version" : "4.0.2",
  "gitVersion" : "fc1573ba18aee42f97a3bb13b67af7d837826b47"
},
"ok" : 1
}
> db.stocks.find({ "stock_volume": { $gt : 176400 } }).count();
2586779
>
```

7.PROGRAMMING ASSIGNMENT

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

```
    },
    {
      "_id" : 6036,
      "value" : {
        "totalMovie" : 888
      }
    },
    {
      "_id" : 6037,
      "value" : {
        "totalMovie" : 202
      }
    },
    {
      "_id" : 6038,
      "value" : {
        "totalMovie" : 20
      }
    },
    {
      "_id" : 6039,
      "value" : {
        "totalMovie" : 123
      }
    },
    {
      "_id" : 6040,
      "value" : {
        "totalMovie" : NaN
      }
    }
  ],
  "timeMillis" : 22132,
  "counts" : {
    "input" : 1000209,
    "emit" : 1000209,
    "reduce" : 6335,
    "output" : 6040
  },
  "ok" : 1
}
>
```

C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe

```
> var mapUser1 = function() {
... emit(this.Gender,1);
... }
> var reduceUser1 = function(key,result){
... return Array.sum(result);
... }
> db
MovieLens
> show collections
rate
ratings
users
> db.users.mapReduce(mapUser1,reduceUser1,{out : {inline : true}});
{
  "results" : [
    {
      "_id" : "F",
      "value" : 1709
    },
    {
      "_id" : "M",
      "value" : 4331
    }
  ],
  "timeMillis" : 114,
  "counts" : {
    "input" : 6040,
    "emit" : 6040,
    "reduce" : 122,
    "output" : 2
  },
  "ok" : 1
}
>
```