Assignment 3 - TDT4225 - Distributed Systems

Group: 29
Students: Lars Ådne Heimdal, Even Flem Hagen
GIT-Repo: github.com/peveneven/distributed_data_assignment3

Introduction

NOTE: Instead of using python we decided to use Golang.

We were given a dataset consiting of recorded outdoor movements by a range of users. The dataset was given in zipfile containing a set of folders and files which had to be parsed and loaded into a MongoDB database. Instead of using python to solve the given tasks, we decided to use GoLang as we were more familiar with the language.

We used Docker to set up a local test enviroment before deploying the code to the VM at NTNU. Furthermore we utilized GIT to collaborate, with the repo being found here. Note that we used a private repository. The repository we have linked to is only a copy, so the commit history is not included.

We decided to split the tasks evenly between us.

Results

Part 1

Users top 10

```
{
        "_id" : "000",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "001",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "002",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "003",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "004",
        "haslabels" : false,
        "activities" : [
                // Omitted to save space!
        ]
}
{
        "_id" : "005",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "006",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "007",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "008",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
{
        "_id" : "009",
        "haslabels" : false,
        "activities" : [
                //omitted to save space!
        ]
}
```

Activities top 10

```
{
    "_id" : ObjectId("5f92b8434500a313d3ea2755"),
    "user_id" : "007",
    "start_date_time" : ISODate("2008-10-25T14:22:00Z"),
    "end_date_time" : ISODate("2008-10-25T14:23:26Z")
}
{
    "_id" : ObjectId("5f92b8444500a313d3ea277c"),
    "user_id" : "007",
    "start_date_time" : ISODate("2008-10-26T16:09:35Z"),
    "end_date_time" : ISODate("2008-10-26T16:17:10Z")
}
{
    "_id" : ObjectId("5f92b8444500a313d3ea27da"),
    "user_id" : "007",
    "start_date_time" : ISODate("2008-10-27T00:56:23Z"),
    "end_date_time" : ISODate("2008-10-27T13:32:11Z")
}
{
    "_id" : ObjectId("5f92b8434500a313d3ea2757"),
    "user_id" : "003",
    "start_date_time" : ISODate("2008-10-23T17:58:54Z"),
    "end_date_time" : ISODate("2008-10-23T18:16:29Z")
}
{
    "_id" : ObjectId("5f92b8434500a313d3ea2756"),
    "user_id" : "006",
    "start_date_time" : ISODate("2008-10-23T06:59:39Z"),
    "end_date_time" : ISODate("2008-10-23T07:22:20Z")
}
{
    "_id" : ObjectId("5f92b8444500a313d3ea2cac"),
    "user_id" : "007",
    "start_date_time" : ISODate("2008-10-28T16:10:31Z"),
    "end_date_time" : ISODate("2008-10-28T16:19:27Z")
}
{
    "_id" : ObjectId("5f92b8444500a313d3ea2cad"),
    "user_id" : "003",
    "start_date_time" : ISODate("2008-10-24T02:02:27Z"),
    "end_date_time" : ISODate("2008-10-24T12:08:47Z")
}
{
    "_id" : ObjectId("5f92b8434500a313d3ea2758"),
    "user_id" : "004",
    "start_date_time" : ISODate("2008-10-23T17:58:52Z"),
    "end_date_time" : ISODate("2008-10-23T18:08:48Z")
}
{
    "_id" : ObjectId("5f92b8444500a313d3ea303b"),
    "user_id" : "007",
    "start_date_time" : ISODate("2008-10-28T16:19:37Z"),
    "end_date_time" : ISODate("2008-10-28T16:27:32Z")
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2761"),
        "user_id" : "011",
        "start_date_time" : ISODate("2008-09-26T10:44:08Z"),
        "end_date_time" : ISODate("2008-09-26T11:04:12Z")
}
```

Trackpoints top 10

```
{
        "_id" : ObjectId("5f92b8434500a313d3ea276d"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345398,
                        39.980202
                ],
                "type" : "Point"
        },
        "altitude" : 175,
        "date_days" : 39746.5986111111,
        "date_time" : ISODate("2008-10-25T14:22:00Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea276e"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345549,
                        39.980547
                ],
                "type" : "Point"
        },
        "altitude" : 186,
        "date_days" : 39746.5986689815,
        "date_time" : ISODate("2008-10-25T14:22:05Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea276f"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345109,
                        39.980585
                ],
                "type" : "Point"
        },
        "altitude" : 10,
        "date_days" : 39746.5990277778,
        "date_time" : ISODate("2008-10-25T14:22:36Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2770"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345264,
                        39.980624
                ],
                "type" : "Point"
        },
        "altitude" : -4,
        "date_days" : 39746.5990856481,
        "date_time" : ISODate("2008-10-25T14:22:41Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2771"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345361,
                        39.980873
                ],
                "type" : "Point"
        },
        "altitude" : 18,
        "date_days" : 39746.5991435185,
        "date_time" : ISODate("2008-10-25T14:22:46Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2772"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345385,
                        39.980883
                ],
                "type" : "Point"
        },
        "altitude" : 59,
        "date_days" : 39746.5992013889,
        "date_time" : ISODate("2008-10-25T14:22:51Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2773"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345385,
                        39.980883
                ],
                "type" : "Point"
        },
        "altitude" : 59,
        "date_days" : 39746.5992361111,
        "date_time" : ISODate("2008-10-25T14:22:54Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2774"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345391,
                        39.980877
                ],
                "type" : "Point"
        },
        "altitude" : 70,
        "date_days" : 39746.5992592593,
        "date_time" : ISODate("2008-10-25T14:22:56Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2775"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345343,
                        39.980866
                ],
                "type" : "Point"
        },
        "altitude" : 84,
        "date_days" : 39746.5993171296,
        "date_time" : ISODate("2008-10-25T14:23:01Z"),
        "user_id" : "007"
}
{
        "_id" : ObjectId("5f92b8434500a313d3ea2776"),
        "activity_id" : ObjectId("5f92b8434500a313d3ea2755"),
        "location" : {
                "coordinates" : [
                        116.345343,
                        39.980866
                ],
                "type" : "Point"
        },
        "altitude" : 84,
        "date_days" : 39746.5993518519,
        "date_time" : ISODate("2008-10-25T14:23:04Z"),
        "user_id" : "007"
}
```

Part 2

1. How many users, activities and trackpoints are there in the dataset (after it is inserted into the database).

```
+------------+---------+
|   TABLE    |  COUNT  |
+------------+---------+
| User       |     182 |
| Activity   |   16046 |
| Trackpoint | 9676756 |
+------------+---------+
```

2. Find the average number of activities per user.

```
Average number of activities per user is: 88.164835
```

3. Find the top 20 users with the highest number of activities.

```
+---------+-------+
| USER ID | COUNT |
+---------+-------+
| "128"   |  2102 |
| "153"   |  1793 |
| "025"   |   715 |
| "163"   |   704 |
| "062"   |   691 |
| "144"   |   563 |
| "041"   |   399 |
| "085"   |   364 |
| "004"   |   346 |
| "140"   |   345 |
| "167"   |   320 |
| "068"   |   280 |
| "017"   |   265 |
| "003"   |   261 |
| "014"   |   236 |
| "126"   |   215 |
| "030"   |   210 |
| "112"   |   208 |
| "011"   |   201 |
| "039"   |   198 |
+---------+-------+
```

4. Find all users who have taken a taxi.

```
+---------+
| USER ID |
+---------+
|     010 |
|     058 |
|     062 |
|     078 |
|     080 |
|     085 |
|     098 |
|     111 |
|     128 |
|     163 |
+---------+
```

5. Find all types of transportation modes and count how many activities that are tagged with these transportation mode labels. Do not count the rows where the mode is null.

```
+----------------+-------+
 | TRANSPORTATION | COUNT |
 +----------------+-------+
 | "walk"         |   481 |
 | "car"          |   419 |
 | "bike"         |   262 |
 | "bus"          |   199 |
 | "subway"       |   133 |
 | "taxi"         |    37 |
 | "airplane"     |     3 |
 | "train"        |     2 |
 | "run"          |     1 |
 | "boat"         |     1 |
 +----------------+-------+
```

6.

a) **Find the year with the most activities.**

The year 2008 had 5894 activities which was the most of any year..

b) **Is this also the year with most recorded hours?**
The year with most activities was not the year with most hours.

7. Find the total distance (in km) walked in 2008, by user with id=112.

```
Distance: 115.474500
```

8. Find the top 20 users who have gained the most altitude meters.

```
+---------+----------+
| USER ID | ALTITUDE |
+---------+----------+
|     062 | 14540m   |
|     085 | 6703m    |
|     084 | 5625m    |
|     128 | 5144m    |
|     078 | 3421m    |
|     112 | 3028m    |
|     082 | 1621m    |
|     073 | 1321m    |
|     021 | 804m     |
|     067 | 718m     |
|     107 | 548m     |
|     153 | 524m     |
|     126 | 316m     |
|     087 | 292m     |
|     163 | 268m     |
|     060 | 213m     |
|     081 | 206m     |
|     086 | 155m     |
|     144 | 130m     |
|     089 | 128m     |
+---------+----------+
```

9. Find all users who have invalid activities, and the number of invalid activities per user

```
+---------+--------------------+
| USER ID | INVALID ACTIVITIES |
+---------+--------------------+
| "000"   |                101 |
| "001"   |                 45 |
| "002"   |                 98 |
| "003"   |                179 |
| "004"   |                219 |
| "005"   |                 44 |
| "006"   |                 17 |
| "007"   |                 30 |
| "008"   |                 16 |
| "009"   |                 31 |
| "010"   |                 50 |
| "011"   |                 32 |
| "012"   |                43 |
| "013"   |                29 |
| "014"   |               118 |
| "015"   |                46 |
| "016"   |                20 |
| "017"   |               129 |
| "018"   |                27 |
| "019"   |                31 |
| "020"   |                20 |
| "021"   |                 7 |
| "022"   |                55 |
| "023"   |                11 |
| "024"   |                27 |
| "025"   |               263 |
| "026"   |                18 |
| "027"   |                 2 |
| "028"   |                36 |
| "029"   |                25 |
| "030"   |               112 |
| "031"   |                 3 |
| "032"   |                12 |
| "033"   |                 2 |
| "034"   |                88 |
| "035"   |                23 |
| "036"   |                34 |
| "037"   |               100 |
| "038"   |                57 |
| "039"   |               147 |
| "040"   |                17 |
| "041"   |               201 |
| "042"   |                54 |
| "043"   |                21 |
| "044"   |                31 |
| "045"   |                 7 |
| "046"   |                13 |
| "047"   |                 6 |
| "048"   |                 1 |
| "050"   |                 8 |
| "051"   |                36 |
| "052"   |                44 |
| "053"   |                 7 |
| "054"   |                 2 |
| "055"   |                15 |
| "056"   |                 7 |
| "057"   |                16 |
| "058"   |                13 |
| "059"   |                 5 |
| "060"   |                 1 |
| "061"   |                12 |
| "062"   |               248 |
| "063"   |                 8 |
| "064"   |                  7 |
| "065"   |                 25 |
| "066"   |                  6 |
| "067"   |                 33 |
| "068"   |                139 |
| "069"   |                  6 |
| "070"   |                  5 |
| "071"   |                 27 |
| "072"   |                  2 |
| "073"   |                 17 |
| "074"   |                 19 |
| "075"   |                  6 |
| "076"   |                  8 |
| "077"   |                  3 |
| "078"   |                 19 |
| "079"   |                  2 |
| "080"   |                  6 |
| "081"   |                 16 |
| "082"   |                 26 |
| "083"   |                 15 |
| "084"   |                 99 |
| "085"   |                182 |
| "086"   |                  5 |
| "087"   |                  3 |
| "088"   |                 11 |
| "089"   |                 40 |
| "090"   |                  3 |
| "091"   |                 63 |
| "092"   |                101 |
| "093"   |                  4 |
| "094"   |                 16 |
| "095"   |                  4 |
| "096"   |                 35 |
| "097"   |                 14 |
| "098"   |                  5 |
| "099"   |                 11 |
| "100"   |                  3 |
| "101"   |                 46 |
| "102"   |                 13 |
| "103"   |                 24 |
| "104"   |                 97 |
| "105"   |                  9 |
| "106"   |                  3 |
| "107"   |                  1 |
| "108"   |                  5 |
| "109"   |                  3 |
| "110"   |                 17 |
| "111"   |                 26 |
| "112"   |                 66 |
| "113"   |                  1 |
| "114"   |                  3 |
| "115"   |                  58 |
| "117"   |                   3 |
| "118"   |                   3 |
| "119"   |                  22 |
| "121"   |                   4 |
| "122"   |                   6 |
| "123"   |                   3 |
| "124"   |                   4 |
| "125"   |                  25 |
| "126"   |                 104 |
| "127"   |                   4 |
| "128"   |                 719 |
| "129"   |                   6 |
| "130"   |                   8 |
| "131"   |                  10 |
| "132"   |                   3 |
| "133"   |                   4 |
| "134"   |                  31 |
| "135"   |                   5 |
| "136"   |                   6 |
| "138"   |                  10 |
| "139"   |                  12 |
| "140"   |                  86 |
| "141"   |                   1 |
| "142"   |                  52 |
| "144"   |                 157 |
| "145"   |                   5 |
| "146"   |                   7 |
| "147"   |                  30 |
| "150"   |                  16 |
| "151"   |                   1 |
| "152"   |                   2 |
| "153"   |                 556 |
| "154"   |                  14 |
| "155"   |                  30 |
| "157"   |                   9 |
| "158"   |                   9 |
| "159"   |                   5 |
| "161"   |                   7 |
| "162"   |                   9 |
| "163"   |                 232 |
| "164"   |                   6 |
| "165"   |                   2 |
| "166"   |                   2 |
| "167"   |                 134 |
| "168"   |                  19 |
| "169"   |                   9 |
| "170"   |                   2 |
| "171"   |                   3 |
| "172"   |                   9 |
| "173"   |                   5 |
| "174"   |                 54 |
| "175"   |                  4 |
| "176"   |                  8 |
| "179"   |                 28 |
| "180"   |                  2 |
| "181"   |                 14 |
+---------+--------------------+
```

10. Find the users who have tracked an activity in the Forbidden City of Beijing

```
+---------+
| USER ID |
+---------+
|     004 |
|     018 |
|     019 |
|     131 |
+---------+
```

11. Find all users who have registered transportation_mode and their most used transportation_mode.

```
+---------+--------+-------+
| USER ID |  MODE  | COUNT |
+---------+--------+-------+
| "010"   | "taxi" |     3 |
| "020"   | "bike" |    80 |
| "021"   | "walk" |     1 |
| "052"   | "bus"  |     1 |
| "056"   | "bike" |    15 |
| "058"   | "car"  |     2 |
| "060"   | "walk" |     1 |
| "062"   | "bus"  |   173 |
| "064"   | "bike" |     1 |
| "065"   | "bike" |    10 |
| "067"   | "walk" |     1 |
| "069"   | "bike" |     1 |
| "073"   | "walk" |    52 |
| "075"   | "walk" |     1 |
| "076"   | "car"  |     3 |
| "078"   | "walk" |    37 |
| "080"   | "taxi" |     1 |
| "081"   | "bike" |     4 |
| "082"   | "walk" |     2 |
| "084"   | "walk" |     9 |
| "085"   | "walk" |    18 |
| "086"   | "car"  |     2 |
| "087"   | "walk" |     5 |
| "089"   | "car"  |     7 |
| "091"   | "walk" |     2 |
| "092"   | "bus"  |     1 |
| "097"   | "bike" |     9 |
| "098"   | "taxi" |     1 |
| "101"   | "car"  |     3 |
| "102"   | "bike" |     7 |
| "107"   | "walk" |     1 |
| "108"   | "walk" |     2 |
| "111"   | "taxi" |     3 |
| "112"   | "walk" |    69 |
| "115"   | "car"  |    80 |
| "117"   | "walk" |     1 |
| "125"   | "bike" |     3 |
| "126"   | "bike" |    13 |
| "128"   | "car"  |   312 |
| "136"   | "walk" |     3 |
| "138"   | "bike" |     2 |
| "139"   | "bike" |     4 |
| "144"   | "walk" |     2 |
| "153"   | "walk" |     5 |
| "161"   | "walk" |     1 |
| "163"   | "bike" |    26 |
| "167"   | "bike" |    31 |
| "175"   | "bus"  |     1 |
+---------+--------+-------+
```

Discussion

Results

Due to a misreading of assignment 2, we ended up with too few activities. This resulted in a lot of "wrong" answers. As such, we are not totally confident that our results from assignment 3 are correct as we cannot confidently compare the results. However, we believe that the number of activites are now correct.

Regarding task 3, 6 and 7, we had different results compared to assignment 3. We find this strange, as the feedback suggested that our answers from assignment 2 was correct.

Finally, we find the results from task 9 a bit strange since it suggests that almost every user has invalid activities. We could not find any errors with our approach, but there might be something we overlooked.

Approach

We decided to have three seperate collections; users, activities and trackpoints. Each activity document has a reference to the user, and each trackpoint document has references to both activity and user. Additionally, each user document has an array containing every activity. As there are max a few thousand activities per user, we found it ok to embed them in the document.

To improve performance for certain queries we created some indexes. Both ***activity_id*** in the trackpoints collection and ***transportation_mode*** in the activities collection where indexes, which helped boost the performance quite a bit on certain queries.

For task 10, we decided to add a geo spatial index for the location field in the trackpoints collection. This makes it quite easy and fast to perform geospatial queries, like finding all users that have tracked an activity at a certain location. specifically, we used the near operator with a 2dsphere index to find trackpoints within a given radius. We were not sure how large of a radius to choose, but ended up with a 100m radius as it gave us the same results as in assignment 2.

We tried to implement task 8 and 9 using mongodb aggregation pipeline, specifically using the accumulator operator. This seemed to work when limiting the amount of trackpoints. However, when performed on the whole dataset the mongoDB server ran out of memory.

Attemted accumulator for task 9:

```
"$accumulator": bson.M{
    "init": `function() {
        return {prevDate: "", invalid: false};
    }`,
    "accumulate": `function(state, date) {
        if (state.prevDate == "") {
            return {prevDate: date, invalid: false};
        }
        if (state.invalid) {
            return state;
        }
        var d1 = Date.parse(state.prevDate);
        var d2 = Date.parse(date);
        return {prevDate: date, invalid: Math.abs(d2 - d1) > 300000};
    }`,
    "accumulateArgs": bson.A{"$date_time"},
    "merge": `function(state1, state2) {
        return state2;
    }`,
    "finalize": `function(state) {
        return state.invalid;
    }`,
    "lang": "js",
},
```

MongoDB vs MySQL

The first thing we noticed was that MongoDB had a much higher write throughput. We also noticed that some queries were slower in mongoDB, particularly those involving joins. We have listed some of the pros and cons we noticed after working with each database:

MongoDB

- **pros**
  - No schema restrictions
  - Easier to follow more advanced queries (due to the pipeline structure)
  - Higher write throughput
- **cons**
  - Group By operations are more difficult
  - Limited join capabilities

MySQL

- **pros**
  - Powerful queries
  - Easier to write simple queries
  - Powerful join capabilities
- **cons**
  - Advanced/long queries can be difficult to read/follow
  - Need to carefully plan table structure

Feedback

It would be helpful to have the correct number of documents we should end up with for each of the tables.