CSC 369: Distributed Computing

Alex Dekhtyar

April 29

Day 11: MongoDB Wrap-up
Transition to MapReduce
Housekeeping

Lab 4: mini-project now
  Test cases
  Will discuss today

- Grading:
  - Way behind.
  - Will prioritize for the rest of the week.
  - Friday lecture might suffer

Lab Period: no separate Zoom today. Office hour in between
Lab 4: How to.

{refresh: true,
  collection: "covid",
  aggregation: "usa",
  time: "month",
  analysis: [{
    task: {track: "positive"},
    output: {
      graph: {
        type: "line",
        legend: "off",
        combo: "combine"
      },
      table: {
        row: "state",
        column: "time",
        title: "COVID cases in the USA this month"
      }
    }
  }]
}
```
Report daily numbers of positive COVID-19 cases from the beginning of the current month through today for the entire US (including territories and possessions)
Lab 4: How to.

Report daily numbers of positive COVID-19 cases from the beginning of the current month through today for the entire US (including territories and possessions)

```javascript
{refresh: true,
collection: "covid",
aggregation: "usa",
time: "month",
analysis: [{
  task: {track: "positive"},
  output: {graph: {
    type: "line",
    legend: "off",
    combo: "combine"},
  table: {
    row: "state",
    column: "time",
    title: "COVID ..."
  }
}]
}

db.covid.aggregate({$match: {date: {$gte: 20200401, $lte: 20200429}}}
    {$project: {_id:0, positive:1, date:1}},
    {$group: {_id:"$date",
               positive: {$sum: "$positive"}}
  }
    {$sort: {date:1}}
)```
Lab 4: How to.

Report daily numbers of positive COVID-19 cases from the beginning of the current month through today for the entire US (including territories and possessions)

db.covid.aggregate({$match: {date: {$gte: 20200401, $lte: 20200429}}})
    {$project: { _id: 0, positive: 1, date: 1 }},
    {$group: { _id: "$date",
        positive: {$sum: "$positive"}
    }},
    {$sort: { date: 1 }},
}
Report daily numbers of positive COVID-19 cases from the beginning of the current month through today for the entire US (including territories and possessions)

```
{refresh: true,
collection: "covid",
aggregation: "usa",
time: "month",
analysis: [{
  task: {track: "positive"},
  output: {graph: {
type: "line",
legend: "off",
combo: "combine"},
table: {
row: "state",
column: "time",
title: "COVID ..."
}}
}
]
```

```
db.covid.aggregate({$match: {date: {$gte: 20200401, $lte: 20200429}}})

{$_id:0, positive:1, date:1},

{$group: {
_id:"$date",
    positive: {$sum: "$positive"}}}

{$sort: {date:1}}
```
General Approach (covid collection)

```javascript
{refresh: true|false,
collection: <collection>,
aggregation: <aggregationLevel>,
time: <timeSpecification>,
target: <states>,
counties: <counties>
analysis: [{task: <taskSpecification>,
    output: <outputSpecification>},
    ...
    {task: <taskSpecification>,
    output: <outputSpecification>}],
Output: <filename>
}
```
### General Approach (covid collection)

```json
{refresh: true|false,
 collection: <collection>,
 aggregation: "50States",
 time: <timeSpecification>,
 target: <states>,
 counties: <counties>
 analysis: [{task: <taskSpecification>,
               output: <outputSpecification>},
            ...
            {task: <taskSpecification>,
               output: <outputSpecification>},
            ],
 Output: <filename>
}
```
General Approach (covid collection)

{refresh: true|false,
 collection: <collection>,
 aggregation: <aggregationLevel>,
 time: <timeSpecification>,
 target: <states>,
 counties: <counties>
 analysis: [{task: <taskSpecification>,
                output: <outputSpecification>},
               ...
               {task: <taskSpecification>,
                output: <outputSpecification>},
               ],
 Output: <filename>
}
General Approach (covid collection)

```json
{refresh: true|false,
collection: <collection>,
aggregation: <aggregationLevel>,
time: <timeSpecification>,
target: <states>,
counties: <counties>
analysis: [{task: <taskSpecification>,
output: <outputSpecification>},
...,{task: <taskSpecification>,
output: <outputSpecification>}],
Output: <filename>
}
```

Select States

Select Time

Project variables

Compute Ratios (project)

Aggregate

Sort
General Approach (covid collection)

{refresh: true|false,
collection: <collection>,
aggregation: <aggregationLevel>,
time: <timeSpecification>,
target: <states>,
counties: <counties>
analysis: [{
    task: <taskSpecification>,
    output: <outputSpecification>
},
...
]
Output: <filename>
General Approach (covid collection)

Select States

Select Time

Project variables

Compute Ratios (project)

Aggregate

Sort

{refresh: true|false,
 collection: <collection>,
 aggregation: <aggregationLevel>,
 time: <timeSpecification>,
 target: <states>,
 counties: <counties>
 analysis: [{task: <taskSpecification>,
  output: <outputSpecification>},
  ...
  {task: <taskSpecification>,
  output: <outputSpecification>},
  ...
  ]
 Output: <filename>
General Approach (covid collection)

Create a data structure template

- Select States
- Select Time
- Project variables
- Compute Ratios (project)
- Aggregate
- Sort
General Approach (covid collection)

- **Select States**
  
  ```
  match: {state: {in: <target>}}
  ```

- **Select Time**
  
  ```
  match: {date: {gte:<start>, lte:<end>}}
  ```

- **Project variables**
  
  ```
  project: {_id:0, <variable>:"$variable"}
  ```

- **Compute Ratios**
  
  ```
  project: {_id:0, ratio: {divide: ["$numerator","$denominator"]}}
  ```

- **Aggregate**
  
  ```
  group: {_id:"$date", <variable>:{$sum: "$variable"}}
  ```

- **Sort**
  
  ```
  sort:{state:1, date:1}
  ```
Switching to Distributed Systems Overview

CSC 469 in 20 minutes
Distributed system

- Multiple autonomous processing nodes

- Network connectivity between them

- Software for coordinating computing activities across autonomous processing nodes
Characteristics

❖ Autonomous components
❖ Different processors, different nodes.
❖ Runs concurrently
❖ Runs asynchronously
❖ Multiple points of control.
❖ Multiple points of failure.
Considerations

- Architecture of control points.
- Distribution of tasks and load balancing.
- Resource sharing between compute nodes.
- The CAP theorem.
- Consistency of data.
- Synchronization.
- Unreachability of resources.
- Communication between nodes.
Benefits

❖ Resource Sharing
❖ Concurrency
❖ Scalability
❖ Fault Tolerance