Big Data

Tutorial #5

Isabelle Kuhlmann
2020-06-12

WeST
People and Knowledge Networks

Outline

1) Lecture Recap
2) Exercise Discussion
3) NoSQL Example: Docker & CouchDB
4) Homework Exercise

What previously happened...

Lecture Recap

NoSQL

- *„...when „NoSQL“ is applied to a database, it refers to an ill-defined set of mostly open-source databases, mostly in the early 21st century, and mostly not using SQL“*
- Some common **characteristics** of NoSQL databases:
  - The relational model is not used
  - Runs well on clusters
  - Open-source
  - Built for the 21st century web technologies
  - Schemaless

Aggregates

- Originate in *domain-driven design*
- Data are stored in *units* which may have a more complex design than simple tuples
- An **aggregate** is a collection of related objects that are supposed to be treated as a unit
  - Thus, data that are commonly accessed or manipulated together, are stored together
  - Application-dependent

Data Model

- Model through which we perceive and manipulate our data (example: relational model)
- Each NoSQL solution has a different data model
- Types of NoSQL data models:
  - Key-value
  - Document
  - Column family
  - Graph
- Focus on aggregate-oriented data models (key-value, document, column-family)
  - They share the notion of aggregates being indexed by a key

Key-Value Stores

- Most basic form of persistent data store
- A key is associated with a an aggregate
- Basically like a map or dictionary
- An aggregate is often simply a string/an array of bytes
- Examples:
  - RockDB
  - LevelDB

Document Stores

- Documents (i.e., aggregates) are indexed by a key
- The aggregate structure is known to the database
- Popular aggregate encodings include JSON and XML
- Ability to query documents beyond the simple lookup of keys
- Examples:
  - CouchDB
  - MongoDB

JSON

- **J**ava**S**cript **O**bject **N**otation
- Open standard file format
- Human-readable
- Data types:
  - **Number:** signed decimal number which may contain a fractional part and may use the E notation
  - **String:** delimited with double quotation marks
  - **Boolean:** either `true` or `false`
  - **Array:** ordered lists of 0 or more values; square bracket notation
  - **Object:**
    - Collection of name-value pairs which are separated by commas
    - Within each pair, name (key) and value are separated by a colon
    - Names are strings and must be unique within an object
    - Delimited with curly brackets
  - **Null:** empty value; using the word `null`

JSON – Example

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

Column Family Stores

- Groups of columns (*column families*) are stored together
  - Each column has to be part of a single column family
  - Assumption: Data of a certain column family will usually be accessed together
- Two-level aggregate structure:
  - 1st level: select a row (through a key)
    - i.e., select an aggregate of interest
  - 2nd level: select a column (again, through a key)
    - Each row is comprised of a map which contains more detailed values, i.e., columns
  - Different rows do not need to contain the same columns!
- Examples: Google Bigtable, Amazon Dynamo, Cassandra, ...

Retail data & communication costs

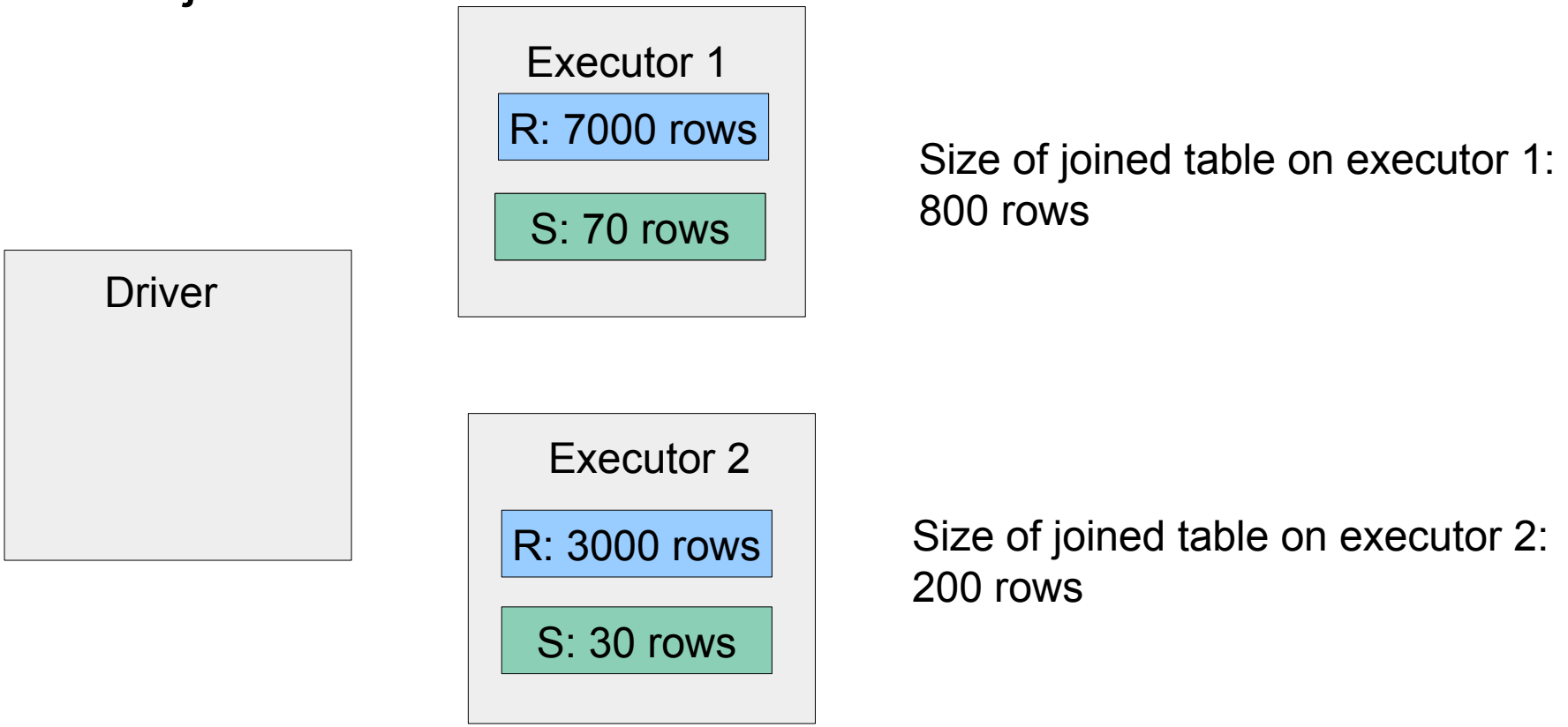
Exercise Discussion

Homework Assignment #1 – Discussion

1) **Pivot**
   Consider the *Retail dataset* again.
   How many instances of each product were sold in each country?
2) **Joins**
   Just play around with the different join types and make sure you understand each one.
   You can utilize `customers.csv` and `orders(2).csv` for this purpose.

Homework Assignment #2 – Discussion

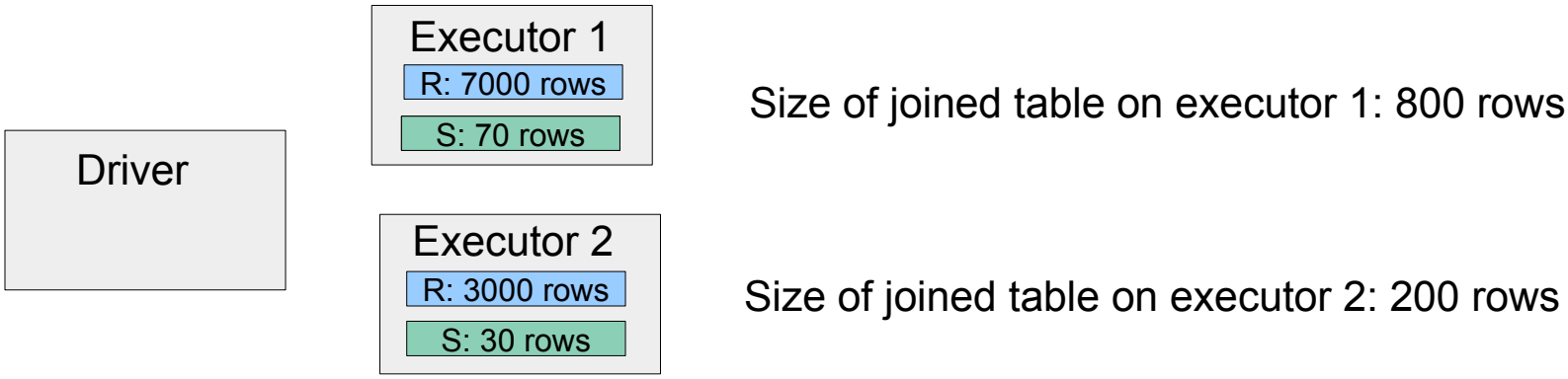
- Compute the **communication cost** and the **elapsed communication cost** for the following big table-to-small table join:



Size of joined table on executor 1: 800 rows

Size of joined table on executor 2: 200 rows

Homework Assignment #2 – Discussion

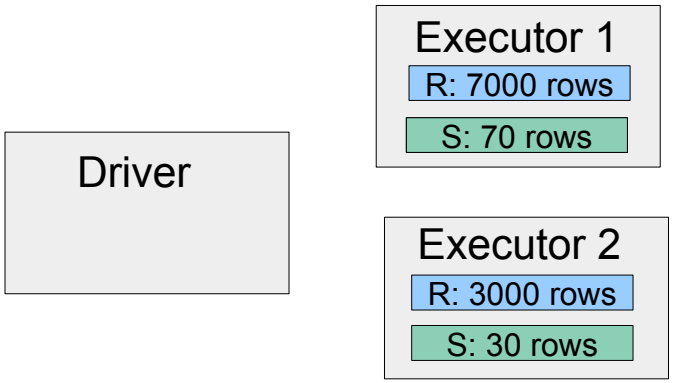


- Communication cost:

  = read parts of S on executors + write S on driver
  + read S on driver + write S on each executor
  + read S on each executor + read parts of R on executors + write outputs

  = (70+30) + (70+30) + 100 + (100+100) + (100+100) + (7000+3000) + (800+200)
  = 700 + 10,000 + 1,000
  = 11,700

Homework Assignment #2 – Discussion



Size of joined table on executor 1: 800 rows

Size of joined table on executor 2: 200 rows

- Elapsed communication cost:
  = max of
- read S on Ex1 + write S on Driver + read S on Driver + write S on Ex1
  + read R on Ex1 + write join result on Ex1 = 70 + 100 + 100 + 100 +7000 + 800 = **8170**
- read S on Ex1 + write S on Driver + read S on Driver + write S on Ex2
  + read R on Ex2 + write join result on Ex2 = 70 + 100 + 100 + 100 + 3000 + 200 = 3570
- read S on Ex2 + write S on Driver + read S on Driver + write S on Ex1
  + read R on Ex1 + write join result on Ex1 = 30 + 100 + 100 + 100 +7000 + 800 = 8130
- read S on Ex2 + write S on Driver + read S on Driver + write S on Ex2
  + read R on Ex2 + write join result on Ex2 = 30 + 100 + 100 + 100 +3000 + 200 = 3530

How to utilize a Docker container to work with CouchDB

NoSQL Example: Docker & CouchDB

Installing Docker and Running CouchDB

- Install Docker
  - Instructions for different platforms can be found here
- Run CouchDB using Docker
  - Here and here you can find some additional information

Now it's your turn!

Homework Exercise

Exercise

- Play around with Docker and a database system of your choice.

Thank you for your Attention!

WeST
People and Knowledge Networks