Big Data

Tutorial #2

Isabelle Kuhlmann
2020-05-08

WeST
People and Knowledge Networks

Outline

1) Lecture Recap
2) Exercise Discussion
3) Spark Introduction (Part 2)
4) Homework Exercise

What previously happened...

Lecture Recap

What is Spark?

- Functional programming
- Lazy transformations & eager actions
- Immutable DataFrames
  - Transformations return new (modified) DataFrames
- Provenance
  - It's always clear where a value comes from and how it has been computed
- On computer clusters

Lineage & Optimization

- Using `.explain()`, we can look at the physical plan Spark created
- Spark internally optimizes this plan
  - Wide transformations are pushed to the end of the execution pipeline
  - *Example:* If we want to filter and sort some data, the filter operation (narrow) is executed before the sort operation (wide)
  - Principle:

    `If`

    $$f \circ g = g \circ f$$

    `And`

    $$efforts(f \circ g) > efforts(g \circ f)$$

    `Then`

    `re-order`

Provenance

- Data is distributed to multiple nodes
  - What to do if one of the nodes fails?
- Distributivity:
  - $f(x \cup y) = f(x) \cup f(y)$
  - $g \circ f(x \cup y) = g(f(x)) \cup g(f(y))$
- Homomorphism:
  - $f(x \cup y) = f(x) \circledast f(y)$
  - $g \circ f(x \cup y) = g(f(x)) \circledast g(f(y))$

Distributivity & Homomorphism – Example

$$\text{count} \circ \text{filter}(\text{df}_1 \cup \text{df}_2 \cup \text{df}_3) =$$

distributivity

$$= \text{count}(\text{filter}(\text{df}_1) \cup \text{filter}(\text{df}_2) \cup \text{filter}(\text{df}_3)) =$$

homomorphism

$$= \text{count}(\text{filter}(\text{df}_1)) + \text{count}(\text{filter}(\text{df}_2) + \text{count}(\text{filter}(\text{df}_3))$$

Word count of *Alice in Wonderland*

Exercise Discussion

Homework Assignment – Discussion

- Calculate the frequency of each word in *Alice in Wonderland* again, but this time, consider the following aspects:
  - Handle capitalized words.
  - Words cannot be empty.
  - Remove punctuation.
  - How do you handle terms like “don’t”, “she’s”, etc.? (Those are actually comprised of two words)
- Demo

More on DataFrames

Spark Introduction (Part 2)

Adding New Columns

- You can use
  - `withColumn(colName, col)` or
  - `select(*cols)`
- Example: Demo

Statistics

- Spark offers a variety of statistical functions, e.g.:
  - Mean
  - Standard deviation
  - Variance
  - Min/max
  - Covariance
  - Correlation
- Example: Demo

Grouping and Aggregating

- Grouping: `groupBy(*cols)`
  - Groups a DataFrame using the specified columns, so we can run aggregations on them
- Aggregation: `agg(*exprs)`
  - The following aggregate functions are available:
    - `avg`
    - `max`
    - `min`
    - `sum`
    - `count`
- Example: Demo

Save Data as .csv File

- The data might be partitioned, and thus, written to several output files
  - Use `coalesce()` to “merge” the data
- Use the write operation to actually write a DataFrame to a file
  - For a .csv file, use `write.format(“csv”)`
- Example: Demo

Now it's your turn!

Homework Exercise

Exercise

- Load the **Retail** dataset, it can be found under `data/retail-data/all/online-retail-dataset.csv` (Link). Answer the following questions/complete the following tasks using Spark:
  - Which item was bought most (total)? Which one was bought most in the USA?
  - Which was the lowest invoice (>0), which one the highest?
  - Add a column which displays whether an item was purchased in Germany.
  - Add a column which shows the total amount of the corresponding invoice.
  - How many German customers spent more than $10?
  - Sort the German customers with respect to their total invoice in descending order.

Thank you for your Attention!

WeST
People and Knowledge Networks