GHENT
UNIVERSITY

imec

FACULTY OF ENGINEERING AND ARCHITECTURE

DEPARTMENT OF INFORMATION TECHNOLOGY (INTEC)
RESEARCH GROUP IDLAB

GEDISTRIBUEERDE GEGEVENSVERWERKING

(E761040)

LAB SESSION 02 - Spark Batch Processing

24/03/2025

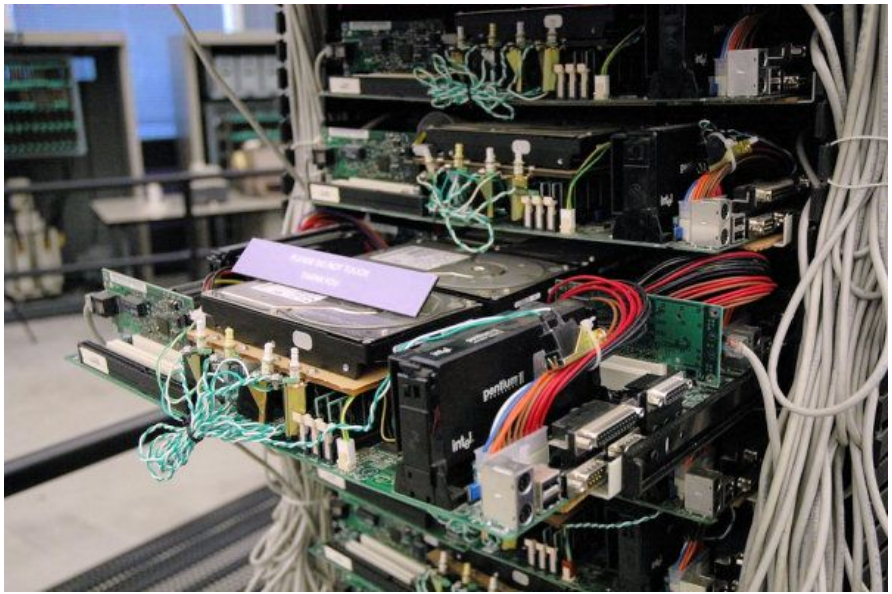
GHENT UNIVERSITY

Introduction

- After EDA, your new task at the video streaming company is to build a recommender system
  - Large datasets with periodic (daily) updates
- MapReduce to the rescue!

MapReduce

- Originated from Google: search engine, analytics
- Key feature: fault tolerance



GHENT UNIVERSITY

MapReduce

- Programming Model to transform data
- Inherently scalable for large workloads
  - Workload spread over many machines simultaneously
- Divide workload based on where the data is
- No complex distributed case in this lab, since the code runs on your PC

Apache Spark (MapReduce)

- Java and Python API
- Distributed execution of batch processing with RDD
- In-memory buffering
- Lazy evaluation
- Much more than just MapReduce!

Lazy Evaluation

Transformations vs Actions

- Transformations change the shape or representation of the data
  - Combined into optimized execution plan
- Execution is only triggered by Actions
  - E.g. Reading rows from transformed dataset

Spark API calls vs notebook code

- Spark API calls run on Spark worker nodes
  - Scalable
  - Optimized execution plan calls Java bytecode
- Regular Python code in between runs in the notebook itself
  - Not scalable
  - Local (interpreted) code

Examples

- Ordering a single column DataFrame
  - Using `sort()`: runs locally in the notebook
  - With `dataframe.orderBy`: spread over all worker nodes
- Both cases in one line:

```
print(avgs.take(10))
```

  - `take(10)` will execute on a Spark node
  - print will execute locally and print the output to stdout

Example Transformations

- `Select`: take a subset of columns
- `Filter`: select rows using a condition
- `Join`: join two DataFrames on a condition
- `Drop`: remove column(s)
- `GroupBy`: group data by column values
- `Sort`: sort rows by one or more columns
- ...

https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql.html#dataframe-apis

Example Actions

- `Take`: take first n rows
- `First`: take first row
- `Collect`: returns array(!) of rows
- `Show`: print the DataFrame's contents (up to 20 rows)
- `Describe`: describe the DataFrame's columns
- ...

https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql.html#dataframe-apis

SQL Functions

Transformations!

- Statistical functions
- Trigonometric functions
- Boolean functions
- Data conversions
- Much, much more

https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql.html#functions

Partitioning

Groups

- `groupBy` to form groups on column
- `agg` performs SQL function on each group
- Single output per group

Windows

- Extensive group manipulation
- Add columns, calculate values per row, ...

https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql.html#window

Lab Session Structure

1. Build a movie recommendation system
   - See https://github.ugent.be/GDV/lab-batch
2. Work on project
   - See https://github.ugent.be/GDV/project-batch
   - Only one dataset per group this time
   - Some groups are assigned a different dataset, be sure to double check!
   - Submit your solution by 30/03/2025 at 23:59

Have fun!
Ask questions.

Lab https://github.ugent.be/GDV/lab-batch

Project https://github.ugent.be/GDV/project-batch

Useful links

- Python Spark API
  https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql.html
- PySpark Cheat Sheet
  https://towardsdatascience.com/ultimate-pyspark-cheat-sheet-7d3938d13421

GHENT UNIVERSITY