SPARK BASICS SIMPLY EXPLAINED

PART 4: SPARK ARCHITECTURE IN ACTION

- *Recap on Sparks architecture*
- *Observe in the code and Spark UI how the main concepts of Spark behave*

Python vs. PySpark – Lazy Evaluation

```python
# Load parquet data from a defined path
pdf = pd.read_parquet(path)

# Filter data
pdf = pdf[pdf["id"] > 10000]

# Count
pdf.shape[0]
```

- Python loads all data in the first line into Memory
- After this the filtering and count is performedc
- For small datasets Python might be faster

```python
# Load parquet data from a defined path
sdf = spark.read.parquet(path)

# Filter data
sdf = sdf.filter(f.col("id") > 10000)

# Count
sdf.count()
```

Transformation

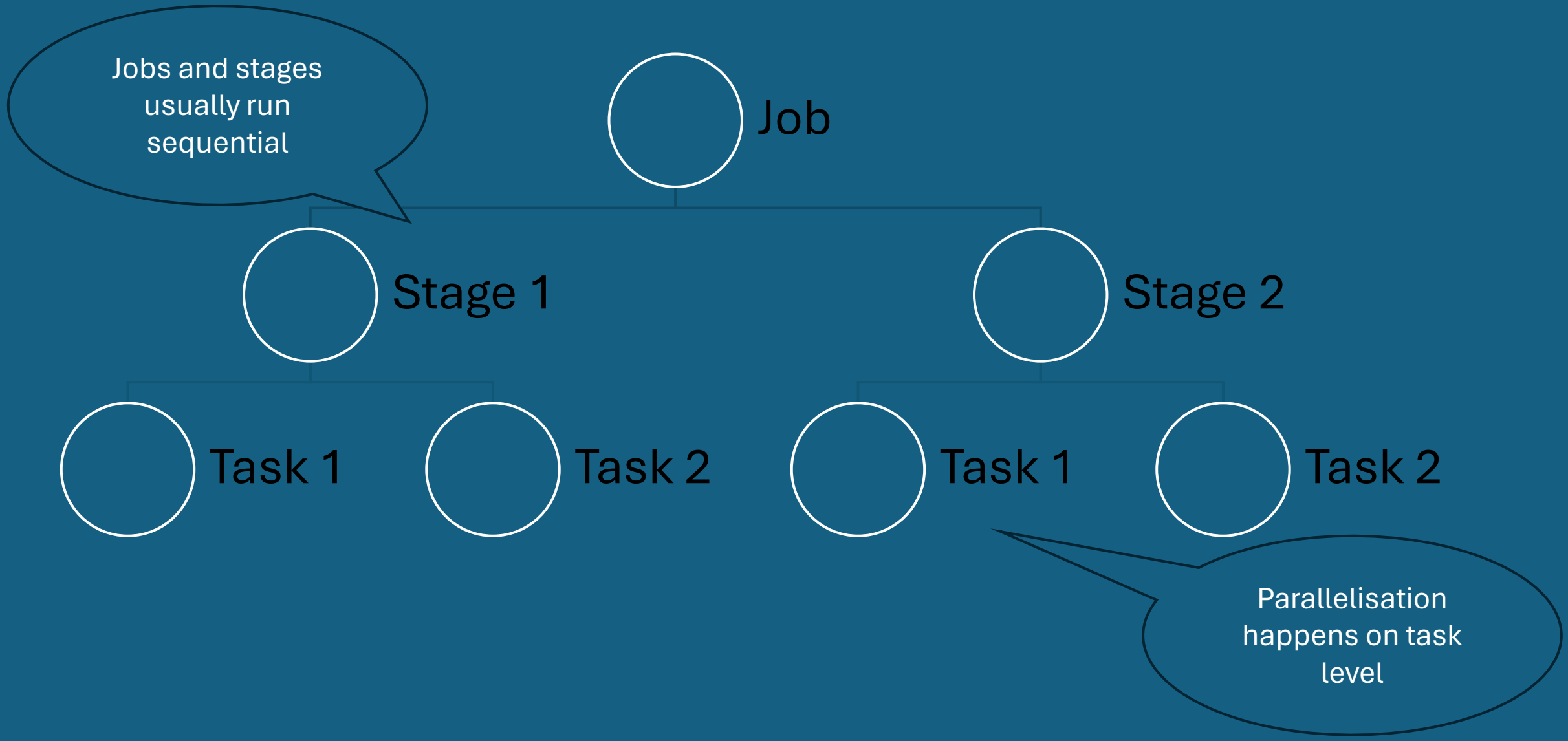
Action

- Spark is loading, filtering and counting data „lazily“ only with the count (called action)
- Based on all identfied steps the „Catalyst Optimizer“ finds the most efficient execution plan

Actions and transformations

- Actions:
  - Actions are methods to access the actual data available in a Dataframe. Action executes all the related transformations to get the required data.
  - Functions such as collect(), show(), count(), first(), take(n) are examples of actions.
- Transformations: Transformations when executed results in a single or multiple new RDD’s.
  - Narrow: Transformations that do not result in data movement between partitions are called Narrow transformations. Examples: select(), union(), filter(), …
  - Wide: Transformations that involves data movement between partitions are called Wide transformations or shuffle transformations. Examples: groupBy(), aggregate(), join(), repartition(), …

Jobs, Stages, Tasks



Jobs, stages, tasks

- Jobs: The highes level in the hierachy consisting of stages and tasks which are distributred across the available cores. One or more jobs are initiated by an action.
- Stages: Jobs are devided into multiple stages. Usually this depends on which operations can be performed in serial or in paralle. Our count example is one where we need two stages. Local and global count. Wide transformations also result into multiple stages
- Tasks: Tasks are the lowest level of work. Each tasks is federated across a core within a worker. Each task executes only one partition. That‘s where the parallisation is happening. If a cluster (driver + worker) has 16 cores then 16 tasks can be executed simultaniasly