Apache Spark Essentials

Quách Đình Hoàng

Contents

Spark Application Concepts

- Application
  - A user program built on Spark using its APIs.
  - It consists of a driver program and executors on the cluster.
- SparkSession
  - An object that provides a point of entry to interact with Spark's APIs.
- Job
  - A parallel computation consisting of multiple tasks
- Stage
  - Each job gets divided into smaller sets of tasks called stages that depend on each other.
- Task
  - A single unit of work or execution that will be sent to a Spark executor.

Spark Application and SparkSession

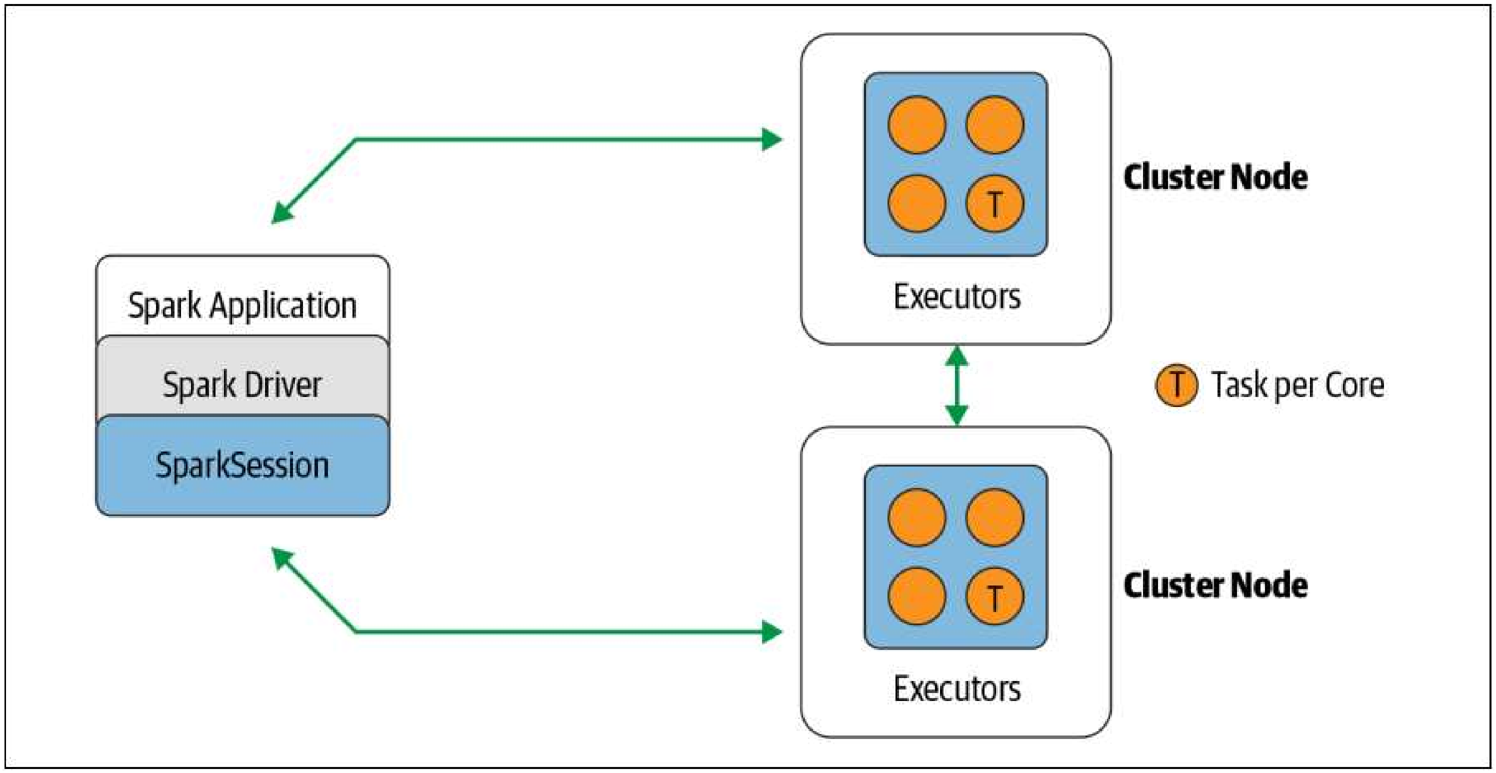


Figure 2-2. Spark components communicate through the Spark driver in Spark's distributed architecture

Spark Jobs

- The driver converts your Spark application into one or more Spark jobs
  - It then transforms each job into a DAG (directed acyclic graph).

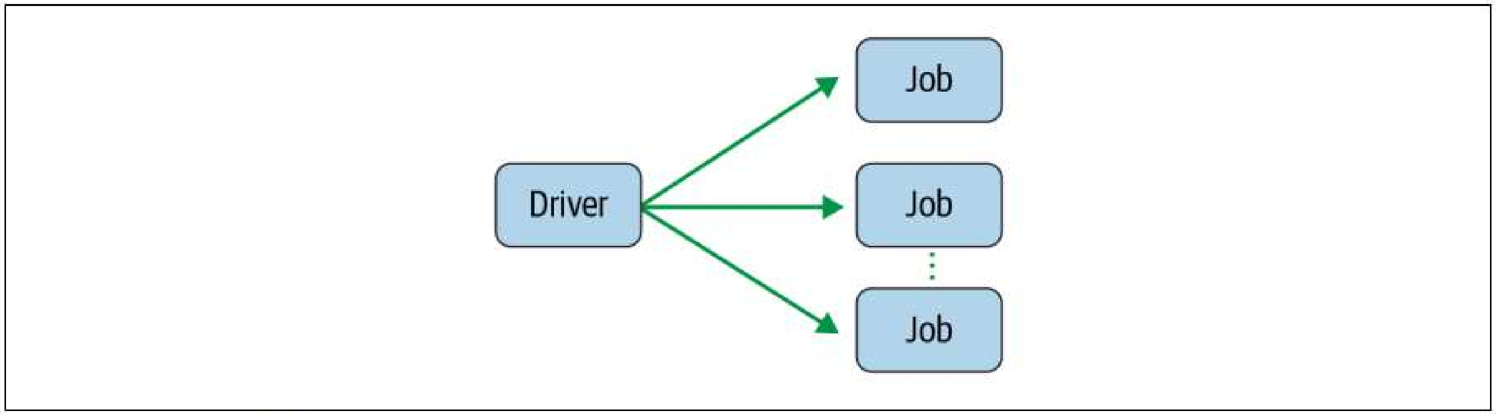


Figure 2-3. Spark driver creating one or more Spark jobs

Spark Stages

- Stages are created based on what operations can be performed serially or in parallel
  - Each Spark operation may be divided into multiple stages

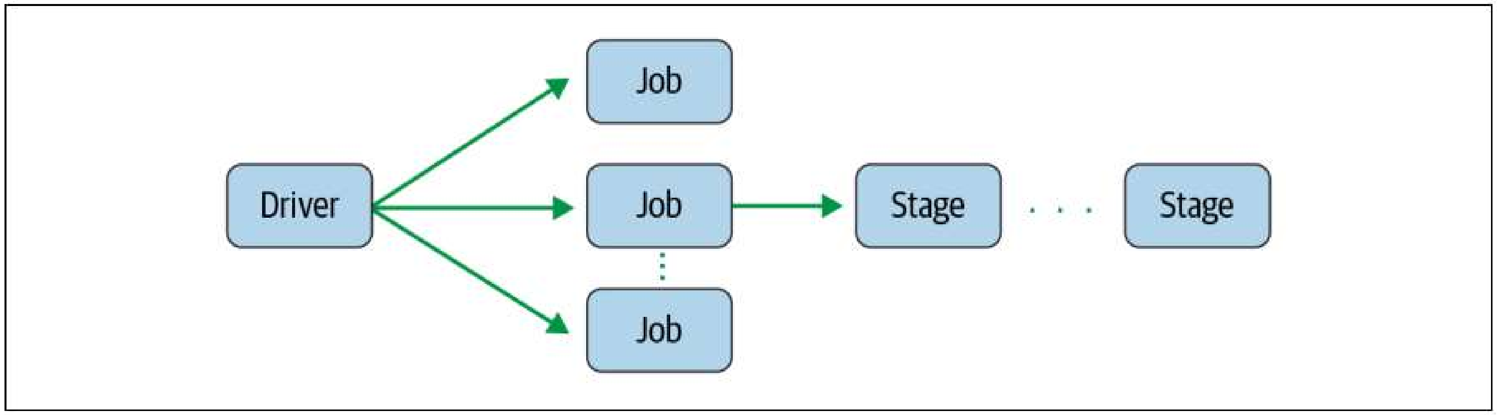


Figure 2-4. Spark job creating one or more stages

Spark Tasks

- Each stage is comprised of Spark tasks (a unit of execution)
  - Each task maps to a single core and works on a single partition of data

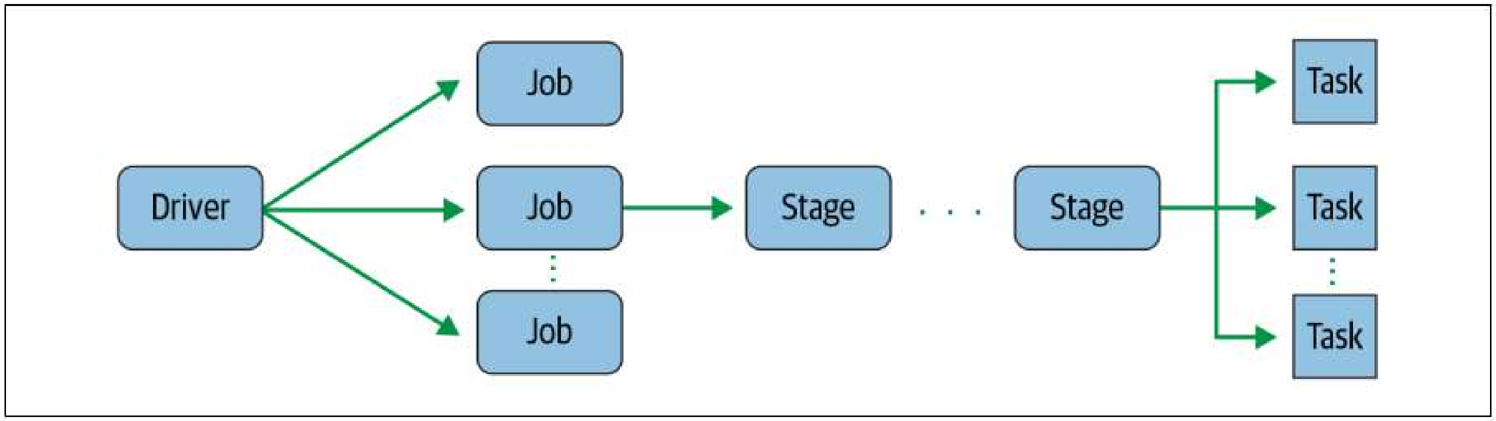


Figure 2-5. Spark stage creating one or more tasks to be distributed to executors

Transformations, Actions, and Lazy Evaluation

- Spark operations on distributed data can be classified into two types: transformations and actions
  - Transformations: transform a Spark DataFrame into a new DataFrame without altering the original data (immutability)
    - All transformations are evaluated lazily (delaying execution until an action is invoked)
    - they are recorded or remembered as a lineage
    - A recorded lineage allows Spark optimize transformations for more efficient execution

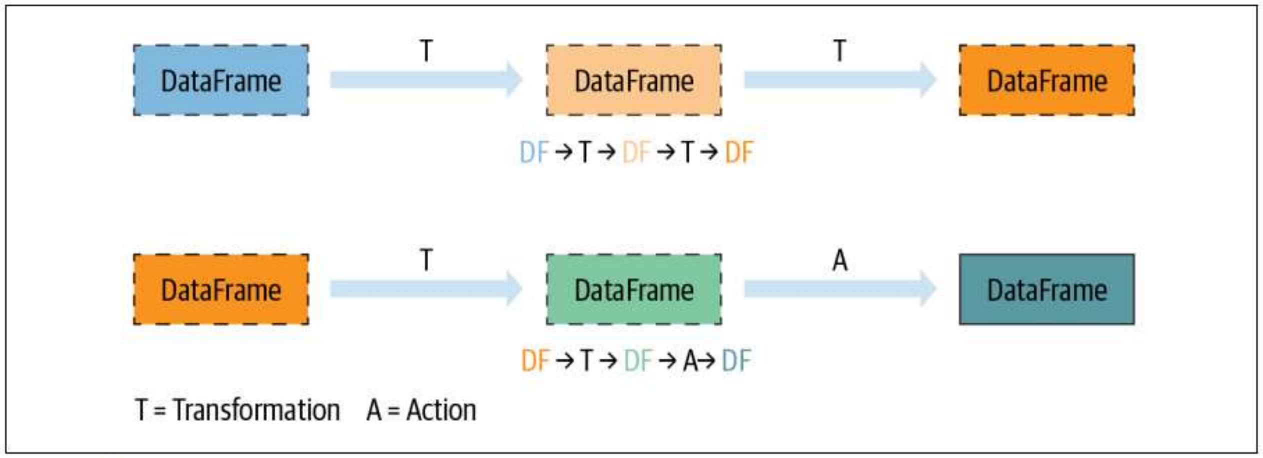
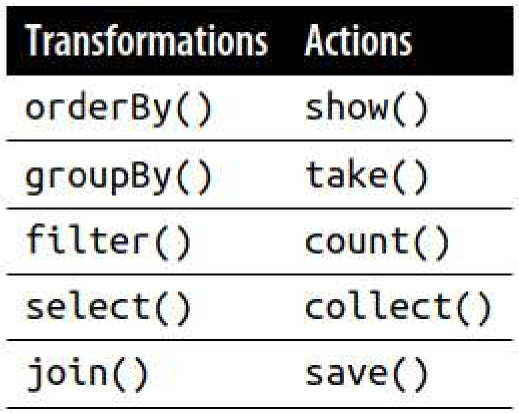


Figure 2-6. Lazy transformations and eager actions

Transformations, Actions, and Lazy Evaluation

- Lazy evaluation allows Spark to optimize transformations (queries)
- Lineage and data immutability provide fault tolerance

Table 2-1. Transformations and actions as Spark operations



Transformations	Actions
`orderBy()`	`show()`
`groupBy()`	`take()`
`filter()`	`count()`
`select()`	`collect()`
`join()`	`save()`

Narrow and Wide Transformations

- Transformations can be classified as having either narrow dependencies or wide dependencies.

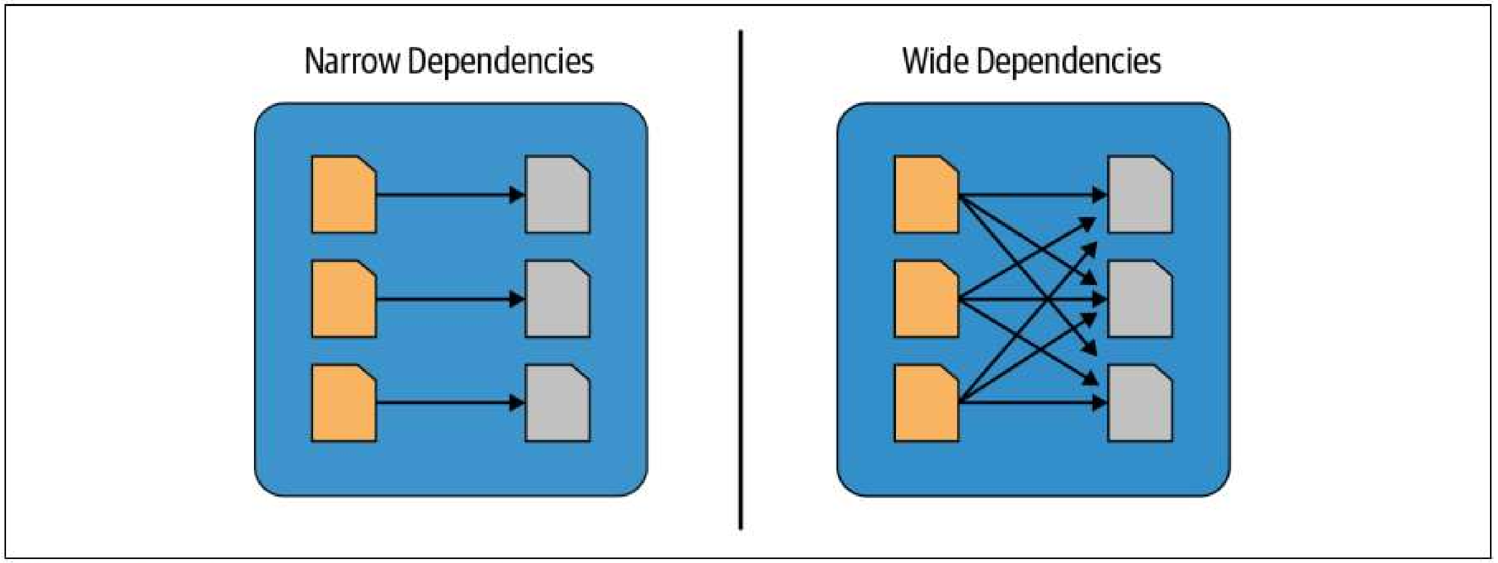


Figure 2-7. Narrow versus wide transformations

The Spark UI

- A graphical user interface can be used to inspect or monitor Spark applications in their various stages of decomposition, that is, jobs, stages, and tasks.
  - A list of scheduler stages and tasks
  - A summary of RDD sizes and memory usage
  - Information about the environment
  - Information about the running executors
  - All the Spark SQL queries

References

- Jules S. Damji, Brooke Wenig, Tathagata Das & Denny Lee, *Learning Spark: Lightning-Fast Data Analytics, 2nd Edition*, O'Reilly, 2020.