Who am I?

- Kevin Jacobs
- Bachelor **Computer Science**
- Currently: Master **Data Science**
Distributed Data Processing

- Significant growth of data
- Single computing node not sufficient
- **Scalable** solutions
- Processing of **data streams**
Streams & Batches

**Streams**
- *Infinite* amount of data

**Batches**
- *Finite* amount of data
Data Processing Frameworks

- **Apache Spark**: “Lightning-fast cluster computing”
- **Apache Flink**: “Scalable Batch and Stream Data Processing”
Flink versus Spark

- Based on **data streams**
- Definition of windows (**time**, **counting**, ...)

- Based on **data batches**
- Batches based on **time**
- Large community
  - More available libraries
Functionality - Comparison

- **Apache Spark**
  - Limited support for *windows*

- **Functionality of Apache Spark**
  - is a subset of *functionality of Apache Flink*
Functionality - What is missing?

- **Metadata driven triggers**
  - Useful for filtering duplicates
My Project

- Is **Apache Flink** better than **Apache Spark**?
  - Stream Enrichment
  - Duplicate Filtering
Stream Enrichment

Stream A

Stateless

Stream B

Stateful

Stream C

... (401, "example-a.org", "CH")
(405, "example-b.org", "NL")
(406, "example-c.org", "DE")
(408, "example-d.org", "CH")
...

... ("CH", 1)
("NL", 2)
("DE", 3)
...

... (401, "example-a.org", "CH", 1)
(405, "example-b.org", "NL", 2)
(406, "example-c.org", "DE", 3)
(408, "example-d.org", "CH", 1)
...
Filtering duplicates

Stream A → Stream B

Stateful
Stateful Stream Processing

Stream A \([\text{Stateless}]\) + Stream B \([\text{Stateful}]\) = Stream C

Stream A \([\text{Stateful}]\) $\rightarrow$ Stream B

- Distributed state:
  - Distributed memory
  - Iterate over elements in state

- Implement the current Spark code in Flink by using distributed memory
Issues - Distributed Memory

What if...

- The memory is full?
  - MemoryStateBackend
  - FsStateBackend
  - ...
Is *Flink* better than *Spark*?

- Compare **Batch API of Flink** with **Batch API of Spark**
  - Benchmark a sorting algorithm on a **fixed-size file**
- Compare **Stream API of Flink** with **Stream API of Spark**
  - Benchmark the **latency** in streaming jobs
Resources

- **Cluster** consisting of 11 nodes
  - Memory: 77GB
  - 44 CPUs (4 per node)
  - 2.5TB HDFS storage
Issues - Cluster

- **Unpredictable resource usage** of the cluster and many other processes running
  - Fixing a small amount of memory for the jobs to prevent memory errors
    - 10 executors, 1GB each
  - Run the experiments a few times to filter out the noise
TeraSort

- **Task:** Sort a fixed-size file (100GB)
- **Input:** HDFS
- **Output:** HDFS
TeraSort - Results (1/3)
TeraSort - Results (2/3)
Execution time:
Apache Flink ≈ ½ Apache Spark

Apache Spark has higher standard deviation for larger jobs
Measuring Latency - Setup (1/2)

- Generate **random strings of bits** of a **fixed length** and append the **creation time** to the bitstring:
  - Example with **size = 2**
    - 10 1470935218352753297
    - 00 1470935218353602960
    - 11 1470935218354432744
    - 00 1470935218355357746
    - 01 1470935218356186778
    - 10 1470935218357042980
    - 00 1470935218357868268
Measuring Latency - Setup (2/2)

- Perform **stateful operations** which keeps **unique bit strings** into memory
  - $2^b$ unique bit strings for bit strings of length $b$

- **Output time - Creation time** $\approx$ **Latency**
Measuring Latency - Result

- There clearly is a winner here!
Conclusion (1/2)

- **Functionality** of Apache Spark is a *subset* of the functionality of Apache Flink

- Apache Flink not yet capable of *mixing* *data streams* and *data batches*
  - But planned for a future version!

- **Tuning is done automatically in Apache Flink**
  - In Apache Spark you need to optimize the parameters yourself
Conclusion (2/2)

● Performance boost
  ○ Apache Flink is faster than Apache Spark in terms of latency and batch processing (at least in the presented use cases)

● Apache Beam combining the good things of both Apache Spark and Apache Flink (in development)

● Use Apache Flink :-)!
Apache Flink

Questions?

mail@kevinjacobs.nl

www.data-blogger.com