CSE 344: Section 8 MapReduce (HW6)

February 22nd, 2018

Apache

Cluster-computing framework

Apache Hadoop Mapreduce vs. Apache Spark

https://www.datamation.com/data-center/hadoop-vs.-spark-the-new-age-of-big-data.html

“Hadoop MapReduce”

Distributed File System (DFS)

MapReduce Job:

- Map Task (EmitIntermediate)
- Reduce Task (Emit)

Fault Tolerance (replicated chunks, write intermediate files to disk)

“Spark” (HW6)

Resilient Distributed Datasets (RDD)

High level commands:

- Transformations (map, join, sort…) -> **Lazy**
- Actions (count, reduce, save...) -> **Eager**

Fault Tolerance (main memory and lineage)

Spark Objects for HW6

```
Row

RowFactory.create(Objects...)

Dataset<Row>

JavaRDD<Row>

JavaPairRDD<K, V>

Tuple2<>
```

you can leave the generics empty

Spark Methods for HW6

```
spark.sql("SELECT ... FROM ...")
```
spark must be a SparkSession

```
d.filter(t -> f(t) == true/false)
```

```
d.distinct()
```

```
d.map()
```
d must be a JavaRDD

```
d.mapToPair(t -> new Tuple2<>(K, V))
```

```
d.reduceByKey((v1, v2) -> f(v1, v2))
```
d must be a JavaPairRDD

MapReduce Framework