Concepts and Models of Parallel and Data-centric Programming

Apache Spark – Programming Model

Lecture, Summer 2020

Simon Schwitanski
Dr. Christian Terboven

i12 High Performance Computing
RWTH AACHEN UNIVERSITY

Outline

0. Organization
1. Foundations
2. Shared Memory
3. GPU Programming
4. Bulk-Synchronous Parallelism
5. Message Passing
6. Distributed Shared Memory
7. Parallel Algorithms
8. Parallel I/O
9. MapReduce
10. **Apache Spark**

a. **Spark Programming Model**
b. Resilient Distributed Datasets (RDDs)
c. Job Scheduling and Fault Tolerance
d. Streaming and Applications
e. Concluding Remarks

Programming Model

- Developers write *Driver* program: Launches workers and assigns tasks
- User code written in Scala, Java or Python
- Workers read from distributed file system (e.g., HDFS) and try to cache computed data in memory
- Main concept: Resilient Distributed Datasets (RDDs)

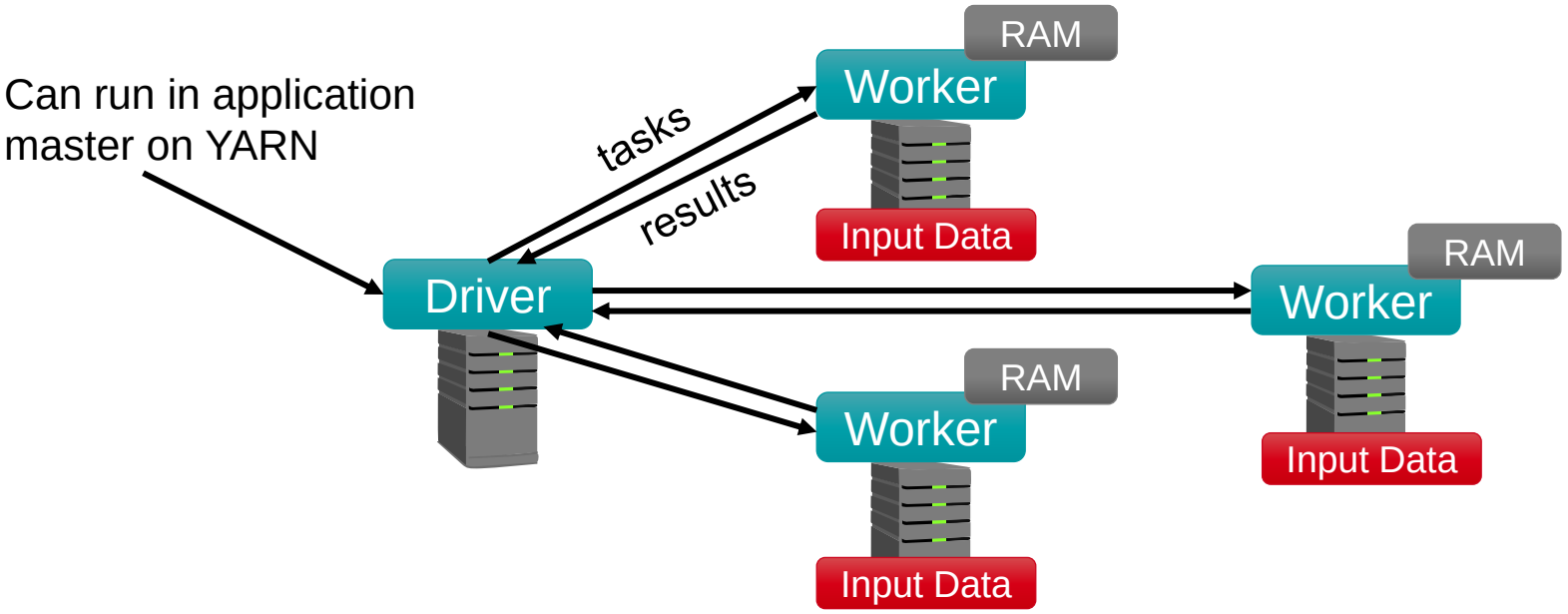
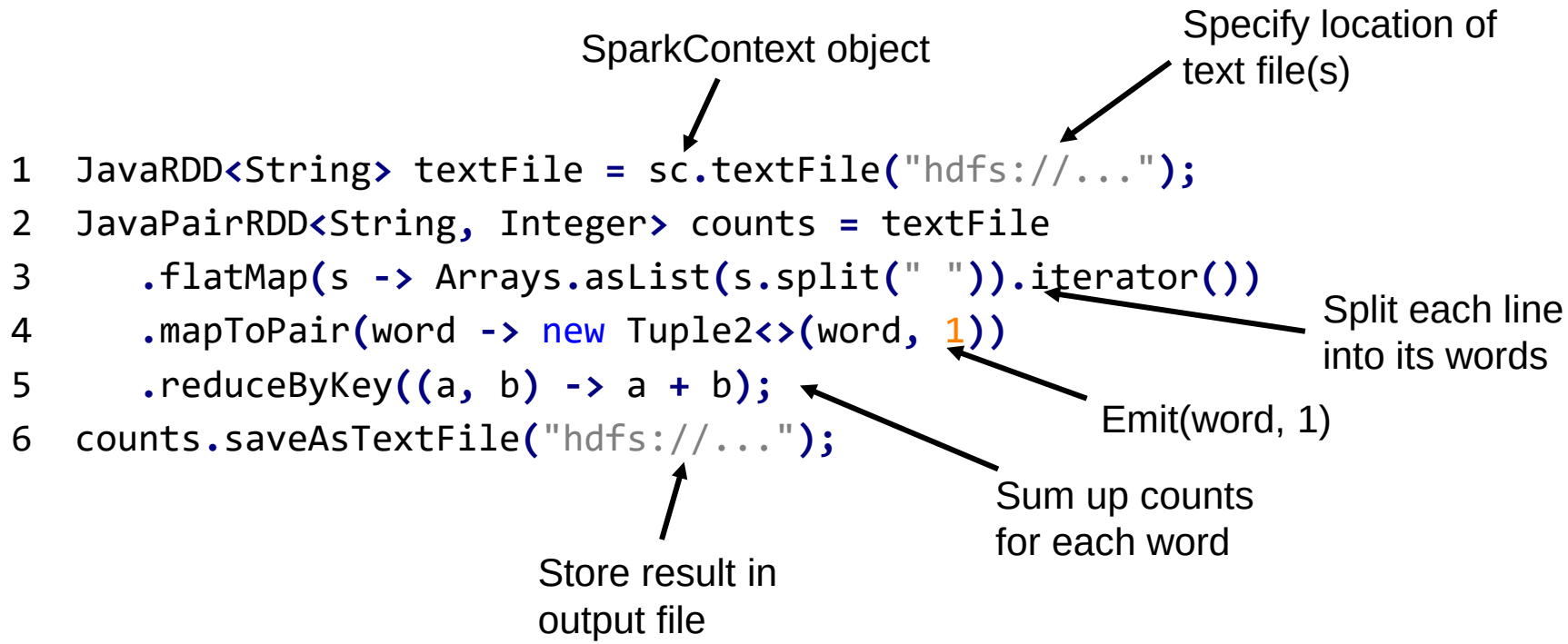


Image Source: Matei Zaharia et al. "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing." NSDI2012: 15-28

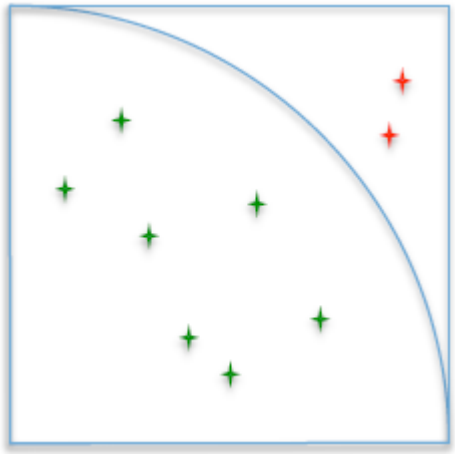
Word Count Example

- Driver program for a word count
  - Load text files in RDD (local file system, HDFS, …)
  - Specify RDD transformations (flatMap, map, reduceByKey)
  - Save resulting RDD in output file



Pi Estimation Example – Java

```
1  final int NUM_SAMPLES = 10000;
2  List<Integer> l = new ArrayList<>(NUM_SAMPLES);
3  for (int i = 0; i < NUM_SAMPLES; i++) {
4    l.add(i);
5  }
6
7  long count = sc.parallelize(l).filter(i -> {
8    double x = Math.random();
9    double y = Math.random();
10   return x*x + y*y < 1;
11 }).count();
12 final double pi = 4.0 * count / NUM_SAMPLES;
```



Pi Estimation Example – Scala

```scala
val count = sc.parallelize(1 to NUM_SAMPLES).filter { _ =>
  val x = math.random
  val y = math.random
  x*x + y*y < 1
}.count()
val x = 4.0 * count / NUM_SAMPLES
```