Steffen Jacobs

22. November 2019

Relevanz und Grundlagen

Was ist Apache Spark?

Wertversprechen

Was kann Apache Spark und wofür ist es zu gebrauchen?

Funktionsweise

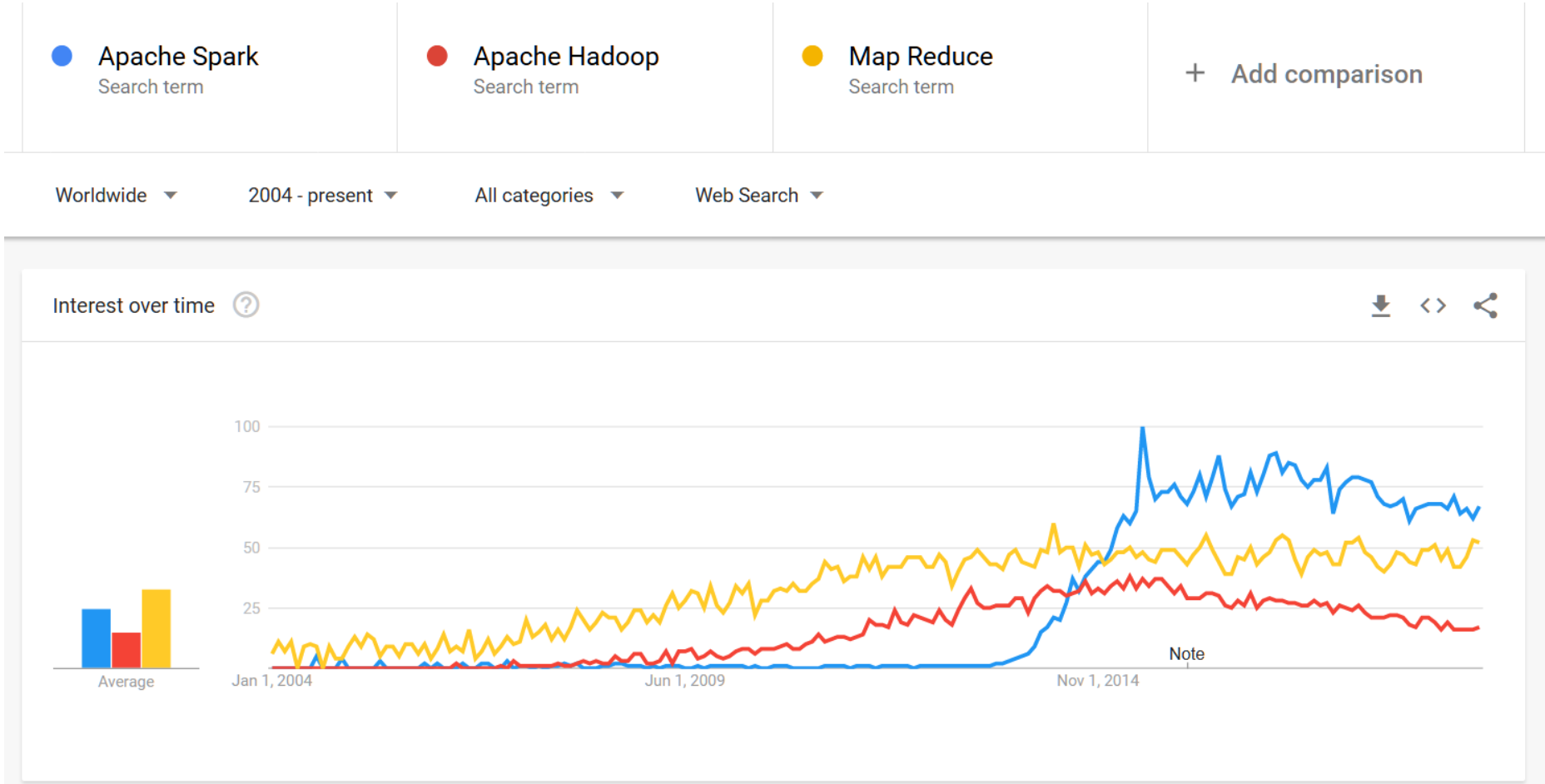
Wie funktioniert Apache Spark?

Demo

Wie funktioniert Apache Spark in der Praxis?

Relevanz

Findet Apache Spark überhaupt Verwendung?



Source: Google Trends

Grundlagen

Was ist Apache Spark?

Datenverarbeitungsengine

Apache Spark ist eine clusterfähige Datenverarbeitungsengine. Alternativen sind Apache Flink, Apache Strom, Akka Reactive Streams, ggf. Hadoop.

In Memory Verarbeitung

Im Kontrast zu Map Reduce auf Hadoop verarbeitet Spark die Daten gleich im Speicher – Das Resultat ist ein significant höherer Durchsatz.

Transformation und Aggregation

Apache Spark soll dabei helfen, große Datenströme möglichst Echtzeit-nah zu verarbeiten.

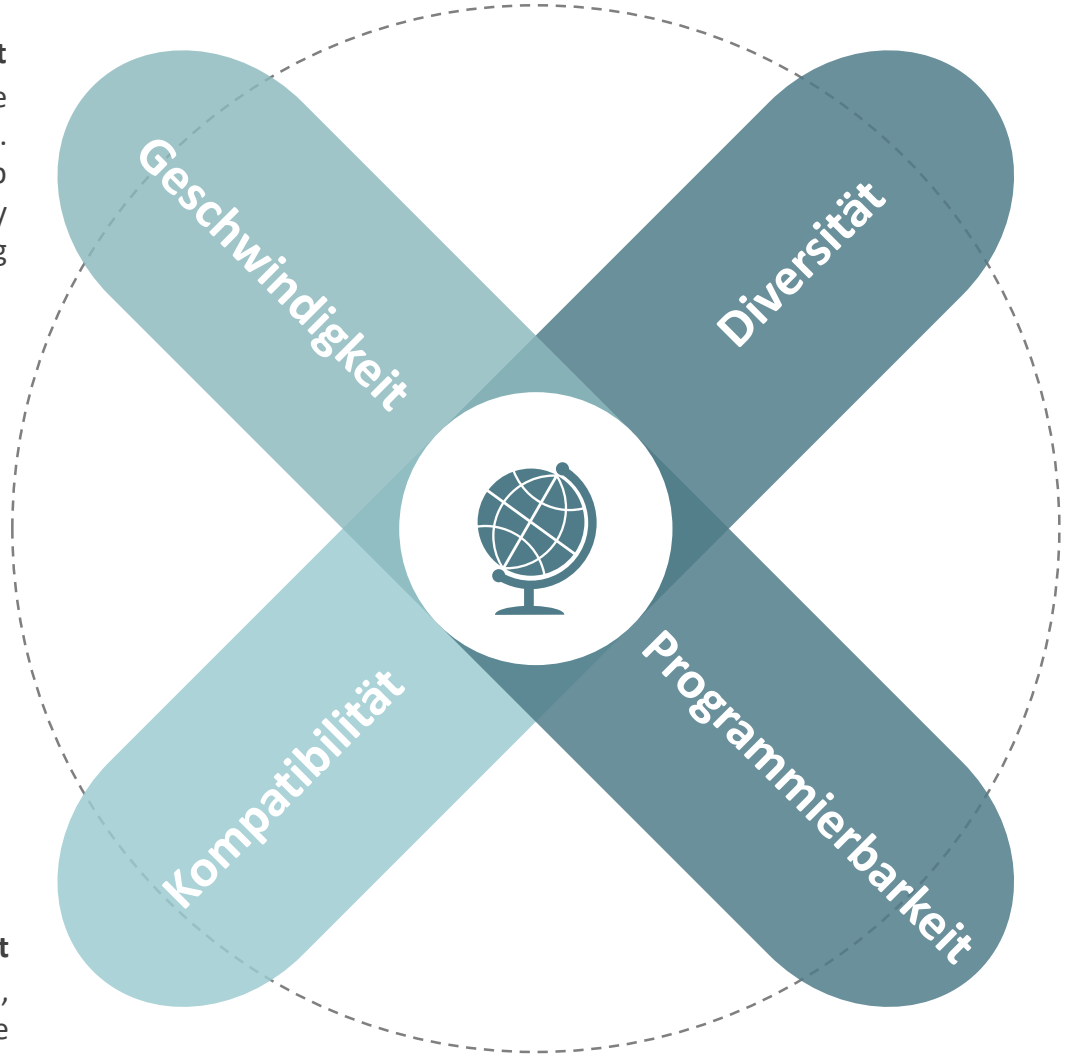
Dataflow Programming

Apache Spark arbeitet direkt mit den Datenitems

Wert-versprechen

Was kann Apache Spark und wofür ist es zu gebrauchen?

Geschwindigkeit
Signifikant bessere Performance als zb. Map-Reduce in Hadoop dank In-Memory Verarbeitung



All-In-One
Spark SQL, Spark Streaming, Mlib (ML), GraphX

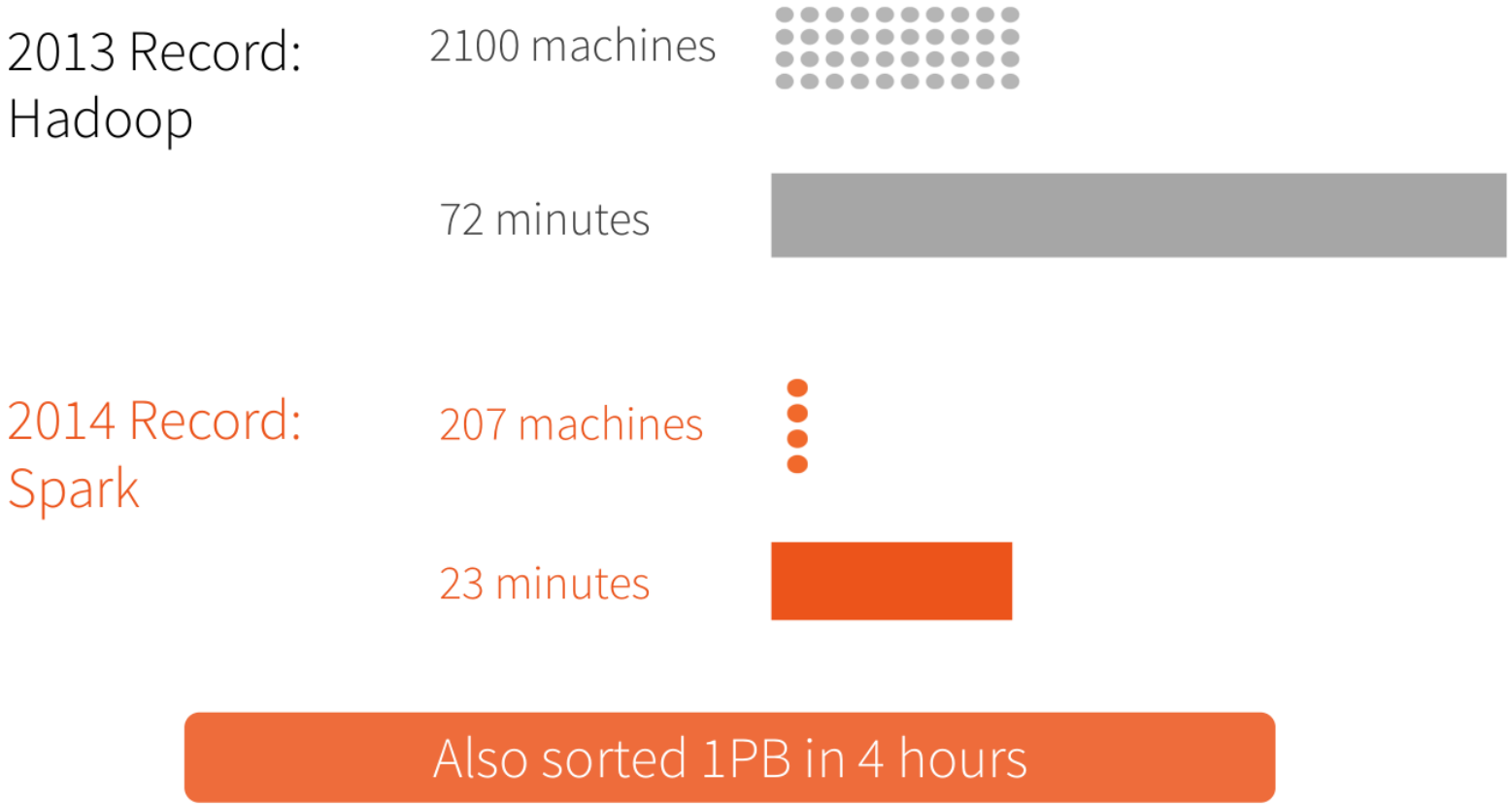
Kompatibel mit
Hadoop, Apache Mesos, Kubernetes, Standalone

Polymorph
Java, Scala, Python, R, SQL, …

Wert-versprechen

Was kann Apache Spark und wofür ist es zu gebrauchen?

Geschwindigkeit 100TB sortieren



Source: Reynold Xin (Stanford)

Wert-versprechen

Was kann Apache Spark und wofür ist es zu gebrauchen?

Programmierbarkeit

```java
package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {

 public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            context.write(word, one);
        }
    }
 }

 public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterable<IntWritable> values, Context context)
      throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
 }

 public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();

        Job job = new Job(conf, "wordcount");

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    job.setMapperClass(Map.class);
    job.setReducerClass(Reduce.class);

    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.waitForCompletion(true);
 }

}
```

Map Reduce: 63 LoC

```scala
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line => line.split(" "))
                 .map(word => (word, 1))
                 .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

Spark: 5 LoC

Source: Rainer Gemulla (Mannheim)

Funktionsweise

Wie funktioniert Apache Spark?

Apache Spark mit Hadoop

Wie funktioniert Apache Spark mit Hadoop?

Apache Spark vs. Map Reduce

Apache Spark oder Map Reduce?

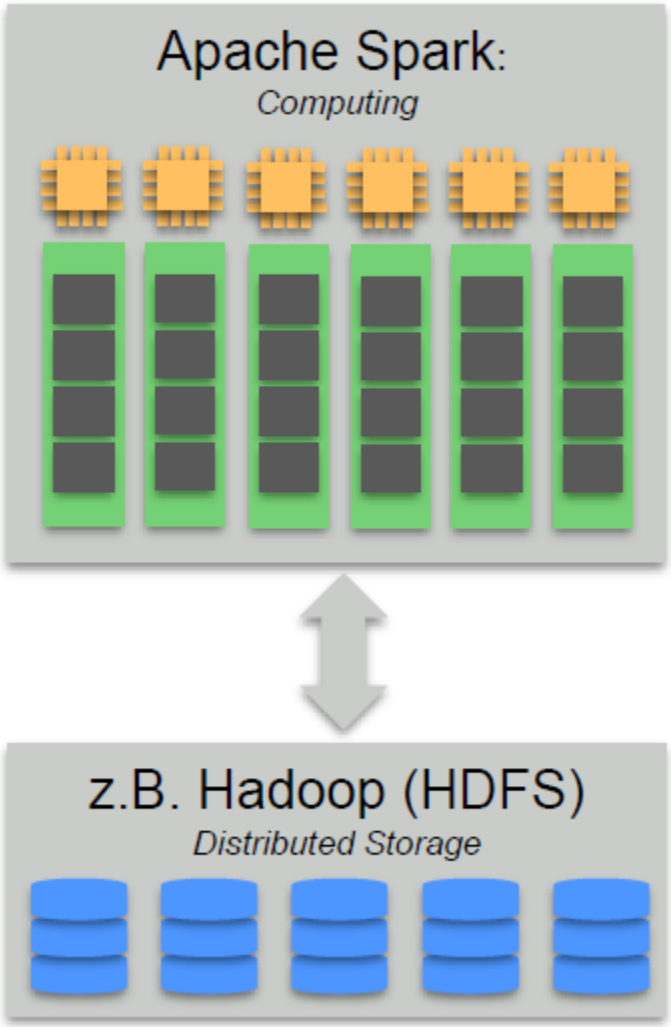
Resilient Distributed Datasets

Die kleinste Dateneinheit in Apache Spark

Funktionsweise

Wie funktioniert Apache Spark?

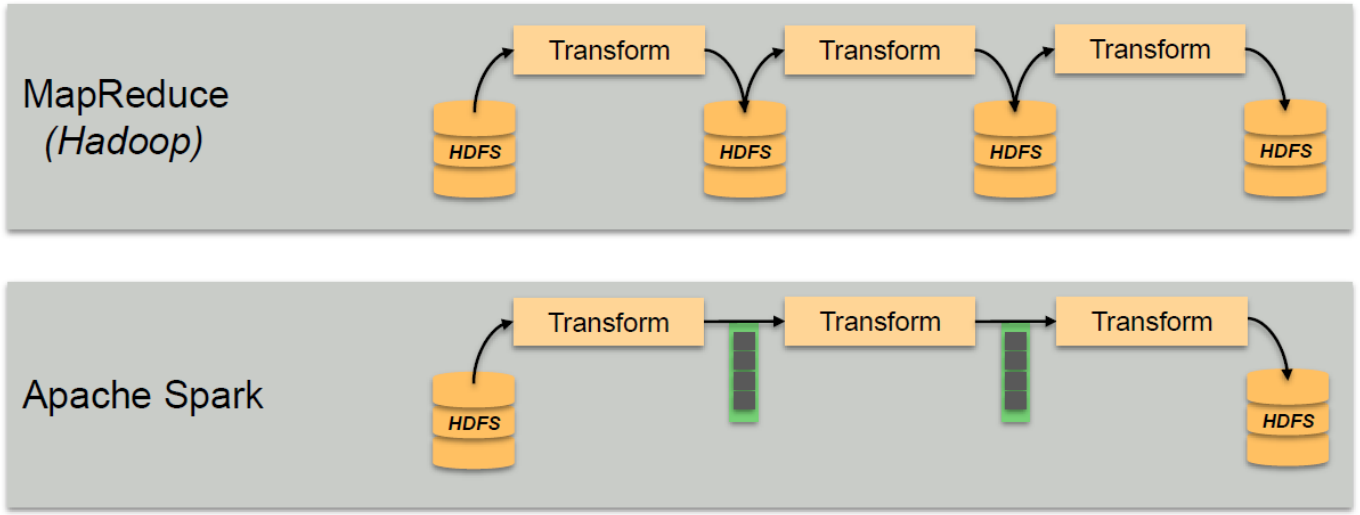
Apache Spark mit Hadoop



Funktionsweise

Wie funktioniert Apache Spark?

Apache Spark
Vs
Map Reduce



Funktionsweise

Wie funktioniert Apache Spark?

Resilient Distributed Datasets

Konzeptionelle Objektmenge

Vergleichbar mit Reihen in einer Datenbanktabelle

Verteilbar

RDDs werden über das Cluster verteilt

Immutable und Disposable

RDDs sind unveränderbar

RDDs speichern keine Daten sondern halten sie nur temporär

Unterstützen Operationen

Transformationen

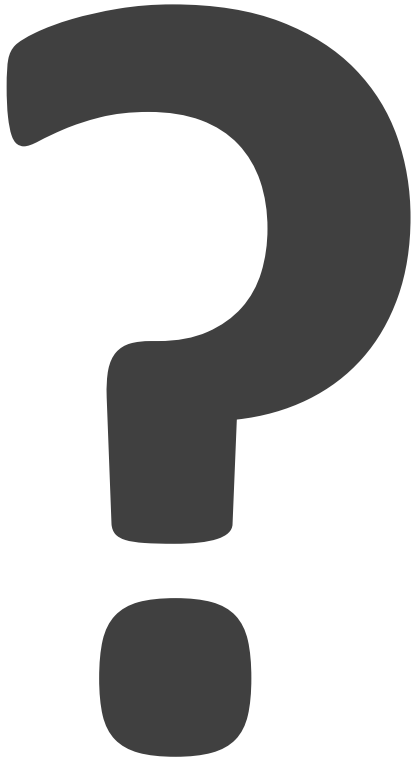
Actions

Demo

Wie funktioniert das in der Praxis?

Fragen

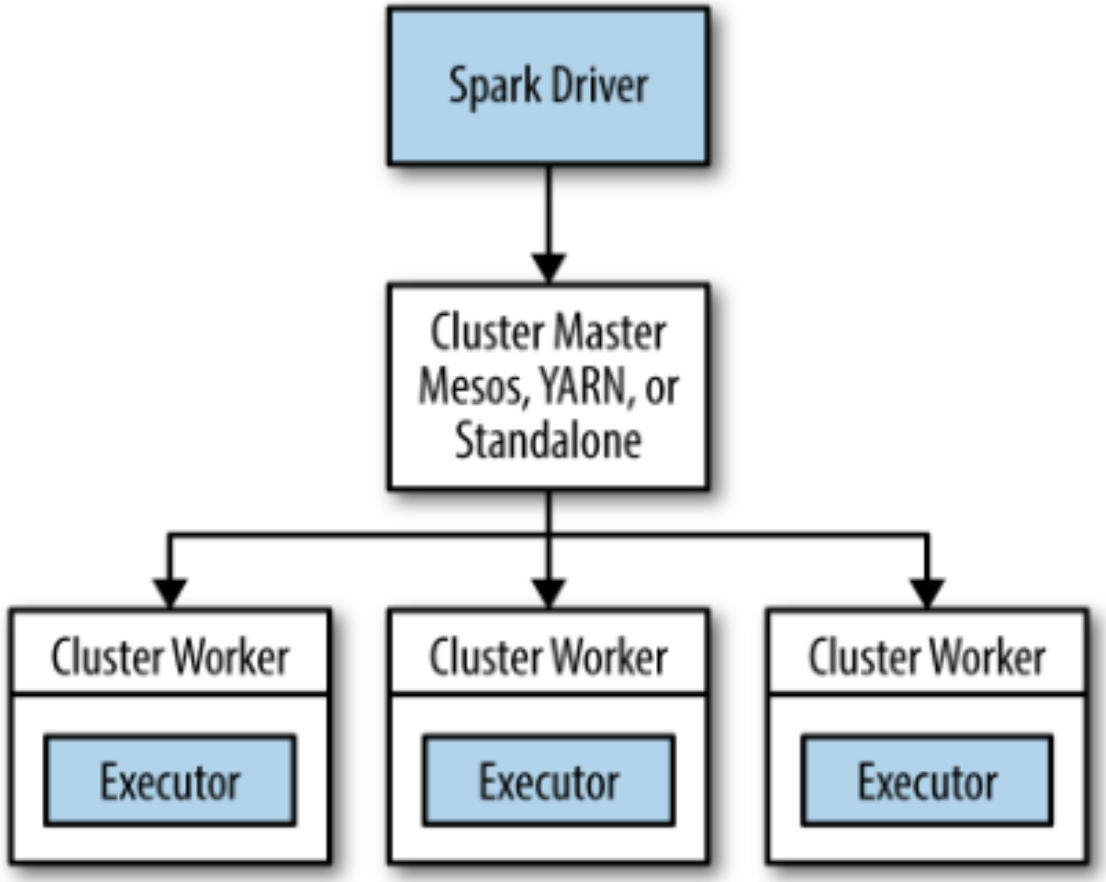
Gibt es Fragen?



Vielen Dank

Backup

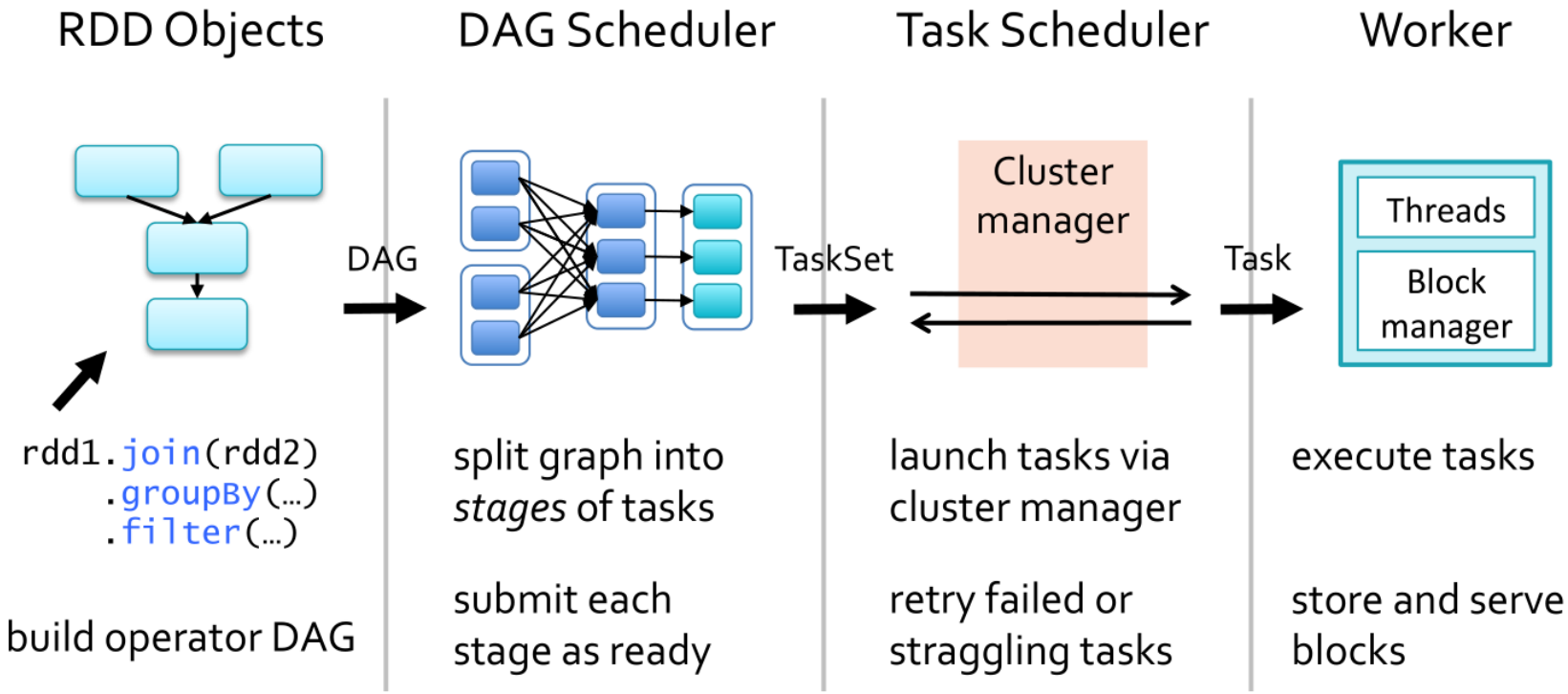
Spark Architektur



Source: Rainer Gemulla (Mannheim)

Backup

Spark Ausführung



Source: Rainer Gemulla (Mannheim)