SparkR

SparkR is an R package that provides a light-weight frontend to use Apache Spark from R

Starting with Spark 1.4.x, SparkR provides a distributed DataFrame implementation that supports operations like selection, filtering, aggregation etc. (similar to R data frames, dplyr) but on large datasets. SparkR also supports distributed machine learning using MLlib.

Hilfreicher Blog bevor Benutzung:

<https://databricks.com/blog/2016/12/28/10-things-i-wish-i-knew-before-using-apache-sparkr.html>

Apache Spark

Apache Spark is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [cluster-computing](https://en.wikipedia.org/wiki/Distributed_computing) [framework](https://en.wikipedia.org/wiki/Software_framework). Originally developed at the [University of California, Berkeley](https://en.wikipedia.org/wiki/UC_Berkeley)'s [AMPLab](https://en.wikipedia.org/wiki/AMPLab), the Spark [codebase](https://en.wikipedia.org/wiki/Codebase) was later donated to the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation), which has maintained it since. Spark provides an [interface](https://en.wikipedia.org/wiki/Application_programming_interface) for programming entire clusters with implicit [data parallelism](https://en.wikipedia.org/wiki/Data_parallelism) and [fault-tolerance](https://en.wikipedia.org/wiki/Fault-tolerant_computing).

If you have Big Data.

Blog der Spark erklärt:

<https://data-science-blog.com/blog/2016/08/03/was-ist-eigentlich-apache-spark/>

Viele Technologieanbieter versprechen schlüsselfertige Lösungen für Big Data Analytics, dabei kann keine proprietäre Software-Lösung an den Umfang und die Mächtigkeit einiger Open Source Projekten heranreichen.

Seit etwa 2010 steht das Open Source Projekt Hadoop, ein Top-Level-Produkt der [Apache Foundation](http://apache.org/), als einzige durch Hardware skalierbare Lösung zur Analyse von strukturierten und auch unstrukturierten Daten. Traditionell im Geschäftsbereich eingesetzte Datenbanken speichern Daten in einem festen Schema ab, das bereits vor dem Laden der Daten definiert sein muss. Dieses Schema-on-Write-Prinzip stellt zwar sicher, dass Datenformate bekannt und –konflikte vermieden werden. Es bedeutet jedoch auch, dass bereits vor dem Abspeichern bekannt sein muss, um welche Daten es sich handelt und ob diese relevant sind. Im Hadoop File System (HDFS) wird ein Schema für erst bei lesenden Zugriff erstellt.

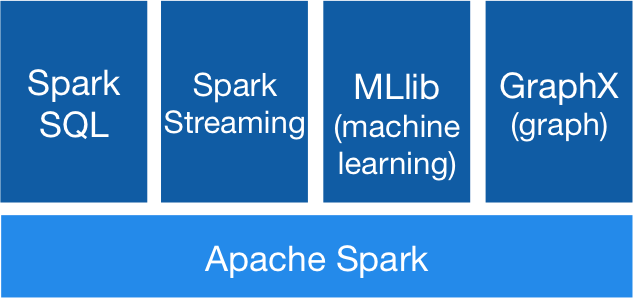
**Apache Spark** ist, ähnlich wie Hadoop, dank Parallelisierung sehr leistungsfähig und umfangreich mit Bibliotheken (z. B. für Machine Learning) und Schnittstellen (z. B. HDFS) ausgestattet. Allerdings ist Apache Spark nicht für jede Big Data Analytics Aufgabe die beste Lösung, Als Einstiegslektüre empfiehlt sich das kostenlose Ebook [Getting Started with Spark: From Inception to Production](http://www.mapr.com/getting-started-apache-spark?source=blog). Wer jedoch erstmal wissen möchte, erfährt nachfolgend die wichtigsten Infos, die es über Apache Spark zu wissen gilt.

**Was ist Apache Spark?**

Apache Spark ist eine Allzweck-Tool zur Datenverarbeitung, eine sogenannte Data Processing Engine. Data Engineers und Data Scientists setzen Spark ein, um äußerst schnelle Datenabfragen (Queries) auf große Datenmengen im Terabyte-Bereich ausführen zu können.

Spark wurde 2013 zum Incubator-Projekt der Apache Software Foundation, eine der weltweit wichtigsten Organisationen für Open Source. Bereits 2014 es wie Hadoop zum Top-Level-Produkt. Aktuell ist Spark eines der bedeutensten Produkte der Apache Software Foundation mit viel Unterstützung von Unternehmen wie etwa Databricks, IBM und Huawei.

**Was ist das Besondere an Spark?**

Mit Spark können Daten transformiert, zu fusioniert und auch sehr mathematische Analysen unterzogen werden.  
Typische Anwendungsszenarien sind interactive Datenabfragen aus verteilten Datenbeständen und Verarbeitung von fließenden Daten (Streaming) von Sensoren oder aus dem Finanzbereich. Die besondere Stärke von Spark ist jedoch das maschinelle Lernen (Machine Learning) mit den Zusätzen MLib (Machine Learning Bibliothek) oder SparkR (R-Bibliotheken direkt unter Spark verwenden), denn im Gegensatz zum MapReduce-Algorithmus von Hadoop, der einen Batch-Prozess darstellt, kann Spark sehr gut iterative Schleifen verarbeiten, die für Machine Learning Algorithmen, z. B. der K-Nearest Neighbor Algorithmus, so wichtig sind.

Spark war von Beginn an darauf ausgelegt, Daten dynamisch im RAM (Arbeitsspeicher) des Server-Clusters zu halten und dort zu verarbeiten. Diese sogenannte In-Memory-Technologie ermöglicht die besonders schnelle Auswertung von Daten. Auch andere Datenbanken, beispielsweise SAP Hana, arbeiten In-Memory, doch Apache Spark kombiniert diese Technik sehr gut mit der Parallelisierung von Arbeitsschritten über ein Cluster und setzt sich somit deutlich von anderen Datenbanken ab. Hadoop ermöglicht über MapReduce zwar ebenfalls eine Prallelisierung, allerdings werden bei jedem Arbeitsschrit Daten von einer Festplatte zu einer anderen Festplatte geschrieben. Im Big Data Umfeld kommen aus Kostengründen überwiegend noch mechanisch arbeitende Magnet-Festplatten zum Einsatz, aber selbst mit zunehmender Verbreitung von sehr viel schnelleren SSD-Festplatten, ist der Arbeitsspeicher hinsichtlich der Zeiten für Zugriff auf und Schreiben von Daten unschlagbar. So berichten Unternehmen, die Spark bereits intensiv einsetzen, von einem 100fachen Geschwindigkeitsvorteil gegenüber Hadoop MapReduce.

Spark kann nicht nur Daten im Terabyte, sondern auch im Petabyte-Bereich analysieren, ein entsprechend großes Cluster, bestehend aus tausenden physikalischer oder virtueller Server, vorausgesetzt. Ähnlich wie auch bei Hadoop, skaliert ein Spark-Cluster mit seiner Größe linear in seiner Leistungsfähigkeit. Spark ist neben Hadoop ein echtes Big Data Framework.  
Spark bringt sehr viele Bibliotheken und APIs mit, ist ferner über die Programmiersprachen Java, Python, R und Scala ansprechbar – das sind ohne Zweifel die im Data Science verbreitetsten Sprachen. Diese Flexibilität und geringe Rüstzeit rechtfertigt den Einsatz von Spark in vielen Projekten. Es kann sehr herausfordernd sein, ein Data Science Team mit den gleichen Programmiersprachen-Skills aufzubauen. In Spark kann mit mehreren Programmiersprachen gearbeitet werden, so dass dieses Problem teilweise umgangen werden kann.

In der Szene wird Spark oftmals als Erweiterung für Apache Hadoop betrachtet, denn es greift nahtlos an HDFS an, das Hadoop Distributed File System. Dank der APIs von Spark, können jedoch auch Daten anderer Systeme abgegriffen werden, z. B. von HBase, Cassandra oder MongoDB.

**Was sind gängige Anwendungsbeispiele für Spark?**

* **ETL / Datenintegration**: Spark und Hadoop eignen sich sehr gut, um Daten aus unterschiedlichen Systemen zu filtern, zu bereinigen und zusammenzuführen.
* **Interaktive Analyse**: Spark eignet sich mit seinen Abfragesystemen fantastisch zur interaktiven Analyse von großen Datenmengen. Typische Fragestellungen kommen aus dem Business Analytics und lauten beispielsweise, welche Quartalszahlen für bestimmte Vertriebsregionen vorliegen, wie hoch die Produktionskapazitäten sind oder welche Lagerreichweite vorhanden ist. Hier muss der Data Scientist nur die richtigen Fragen stellen und Spark liefert die passenden Antworten.
* **Echtzeit-Analyse von Datenströmen**: Anfangs vor allem zur Analyse von Server-Logs eingesetzt, werden mit Spark heute auch Massen von Maschinen- und Finanzdaten im Sekundentakt ausgewertet. Während Data Stream Processing für Hadoop noch kaum möglich war, ist dies für Spark ein gängiges Einsatzgebiet. Daten, die simultan von mehreren Systemen generiert werden, können mit Spark problemlos in hoher Geschwindigkeit zusammengeführt und analysiert werden.  
  In der Finanzwelt setzen beispielsweise Kreditkarten-Unternehmen Spark ein, um Finanztransaktionen in (nahezu) Echtzeit zu analysieren und als potenziellen Kreditkartenmissbrauch zu erkennen.
* **Maschinelles Lernen**: Maschinelles Lernen (ML – Machine Learning) funktioniert desto besser, je mehr Daten in die ML-Algorithmen einbezogen werden. ML-Algorithmen haben in der Regel jedoch eine intensive, vom Data Scientist betreute, Trainingsphase, die dem Cluster viele Iterationen an Arbeitsschritten auf die großen Datenmengen abverlangen. Die Fähigkeit, Iterationen auf Daten im Arbeitsspeicher, parallelisiert in einem Cluster, durchführen zu können, macht Spark zurzeit zu dem wichtigsten Machine Learning Framework überhaupt.  
  Konkret laufen die meisten Empfehlungssysteme (beispielsweise von Amazon) auf Apache Spark.