

Schneller als Hadoop? Einführung in Spark Cluster Computing

Dirk Reinemann
Oracle Deutschland B.V. & Co. KG
Potsdam

Schlüsselworte

Big Data, Cluster Computing, Spark, Shark, Resilient Distributed Datasets, Apache Hadoop

Einleitung

Diese Präsentation wird sich mit dem Thema der Verarbeitung von großen Datenmengen auseinandersetzen. Dieses allgegenwärtige Thema erlangt eine immer größer werdende Bedeutung. Dabei gibt es viele Lösungen welche sich in den letzten Jahren mit mehr oder weniger großem Erfolg auf dem Markt durchsetzen konnten. Eines der erfolgreichsten Lösungen ist Apache Hadoop. Dieses auf dem von Google entwickeltem Programmiermodell MapReduce basierendem Framework verarbeitet große anfallende Datenmengen in zwei Schritten - Map und Reduce.

Dabei kann es zu einigen Problemen beim Einsatz von Apache Hadoop kommen. Der dauerhaft stattfindende Zugriff auf das Dateisystem kann zu Engpässen in der Anwendung führen oder einen sehr großen Ressourcenverbrauch produzieren. Auch die nicht vorhandene Wiederverwendung der Daten kann eine Auswirkung auf die Systemlast haben. Die gesammelten Daten werden in verschiedenen Schritten und Zyklen unabhängig voneinander verarbeitet und können nicht übergreifend zwischen den verschiedenen Instanzen genutzt werden. Hinzu kommt die Einschränkung auf das MapReduce Programmiermodell und die dabei entstehende Lücke zu speziellen Anwendungsfällen in denen ein flexibleres Programmiermodell gefordert wird.

Diese Schwächen versucht das von der UC Berkeley entwickelte Spark Cluster Computing System mit seiner fehlertoleranten Abstraktion für In-Memory Technologie, den Resilient Distributed Datasets, zu lösen. Es bietet ein sehr flexibles Programmiermodell für unterschiedlichste Anwendungsgebiete und kann daher in den verschiedensten Szenarios eingesetzt werden. Außerdem besteht die Möglichkeit der Speicherung von Daten in einem von allen Instanzen nutzbaren Speicherbereiche. Bereits berechnete Daten werden dort abgelegt, um eine Neuberechnung beim nächsten Durchlauf zu vermeiden. Dadurch werden Ressourcen gespart und viele Verarbeitungsfälle um ein vielfaches schneller ausgeführt als es bei Apache Hadoop der Fall ist.

Motivation

Der Abschnitt Motivation beschreibt das Ausgangsproblem und die Schwächen aktueller Systeme zur Verarbeitung großer Datenmengen. Dabei wird auf die bereits angesprochene Wiederverwendung von Daten und der dauerhafte Zugriff auf das Dateisystem eingegangen, welche zu hohem Ressourcenverbrauch der Anwendung führen können. Anschließend wird ein kurzer Überblick der Resilient Distributed Datasets gegeben und deren Merkmale erläutert.

von Festplatte
lesen

auf Festplatte
schreiben

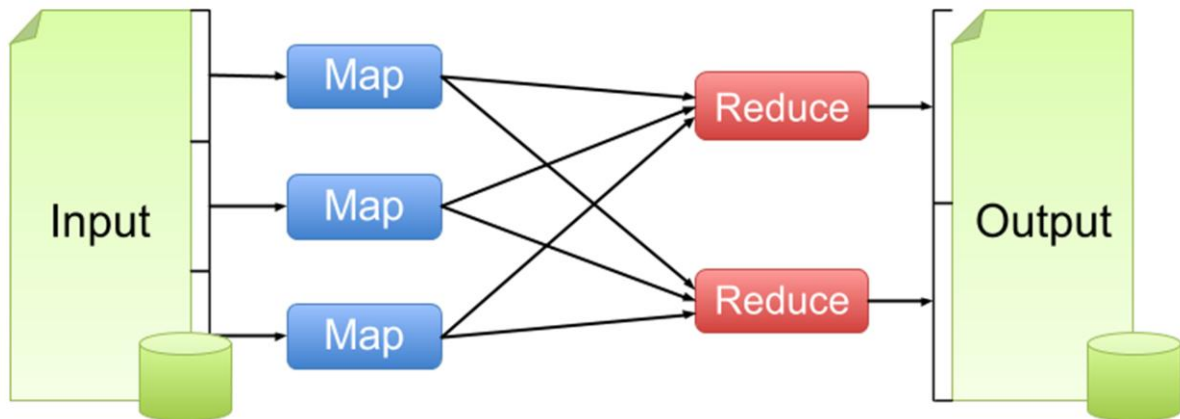
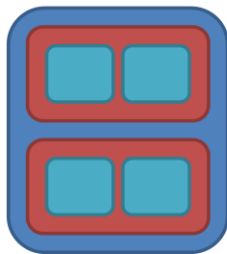


Abbildung 1: Ausgangsproblem

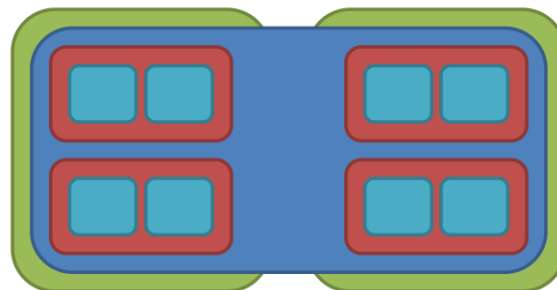
Infrastruktur

Der Abschnitt Infrastruktur wird zunächst den Aufbau der Resilient Distributed Datasets im Detail dargelegt und Begrifflichkeiten für den weiteren Verlauf der Präsentation klären.

RDD



RDD im Cluster



Datensatz Partition RDD Clusterknoten

Abbildung 2: Aufbau der Resilient Distributed Datasets

Danach wird auf die zwei Arten von Operationen eingegangen, welche auf dem System erfolgen können und jeweilige Beispiele sowie deren Ein- und Ausgabewerte erläutert. Nachdem der Besucher eine Vorstellung von der Funktionsweise des Systems erhält, wird das Thema Ausfallsicherheit zuerst schematisch und anschließend an einem konkreten Beispiel aufgezeigt. Dieses wird die Auswertung von Logdateien sein.

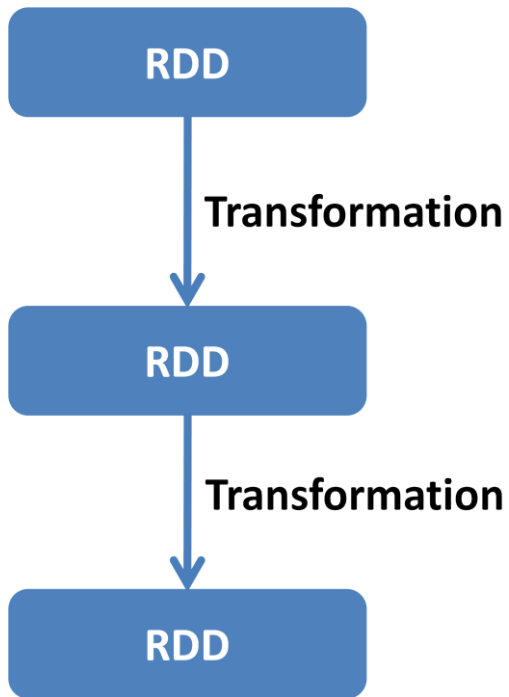


Abbildung 3: Lineage Graph

Anschließend folgt die Darstellung der verschiedenen Abhängigkeiten anhand mehrerer Operationen und grafischen Darstellungen, welche die einzelnen Instanzen zueinander haben können.

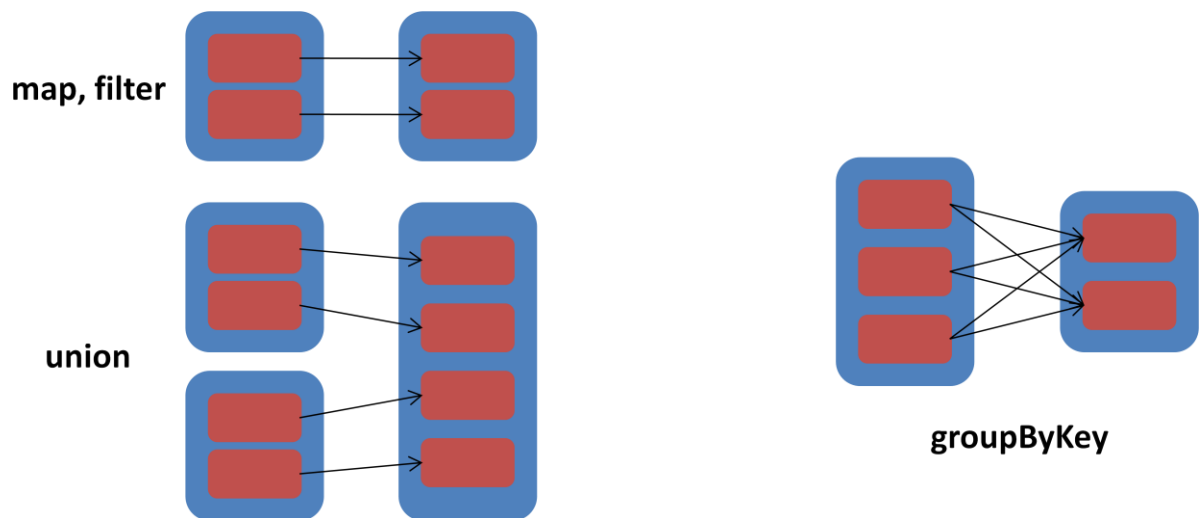


Abbildung 4: Narrow And Wide Dependencies

Der letzte Unterpunkt ist das Feintuning des Systems durch den Entwickler. Der Besucher erhält einen Eindruck darüber wie flexibel das Programmiermodell ist und unter welchen Gesichtspunkten das System kontrolliert werden kann.

Performance

Der Abschnitt Performance beschäftigt sich mit der Performance der Resilient Distributed Datasets. Dabei wird der Vergleich zum Apache Hadoop eine ausschlaggebende Rolle spielen. Anhand vieler Grafiken und Anwendungsfälle wird aufgezeigt, dass Spark durch die Wiederverwendung von Daten und ausbleibenden Dateisystemzugriffen eine teilweise erhebliche Steigerung der Verarbeitungsgeschwindigkeit erreicht.

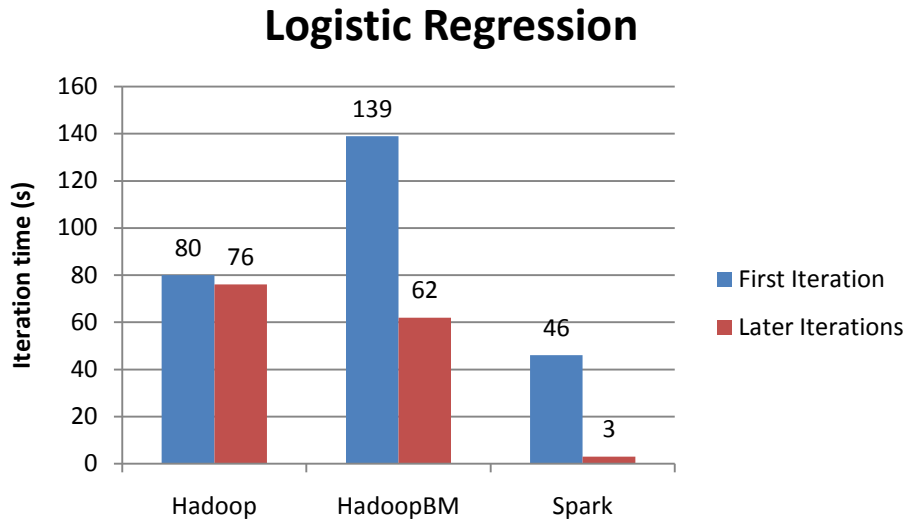


Abbildung 5: Verarbeitungsgeschwindigkeit von Logistic Regression verschiedener Iterationen im Vergleich

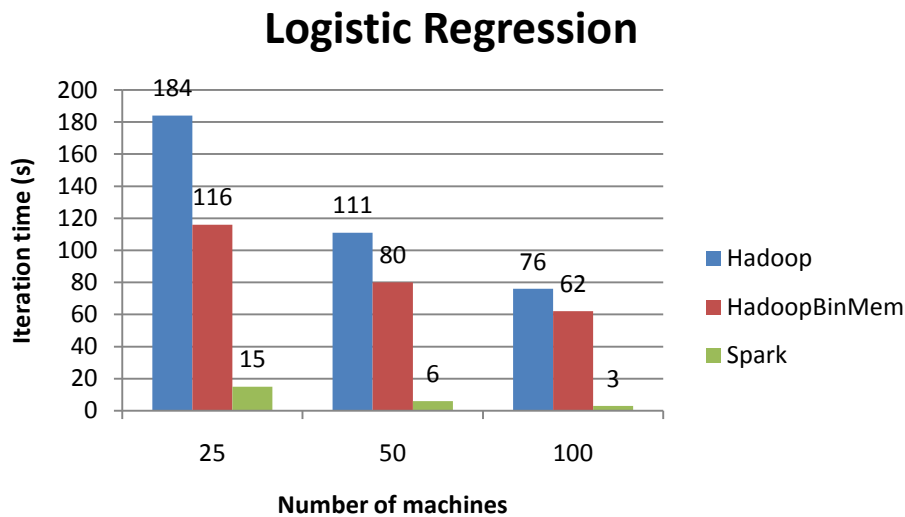


Abbildung 6: Verarbeitungsgeschwindigkeit von Logistic Regression verschiedener Anzahl an Maschinen im Vergleich

Performance of logistic regression using 100 GB data on 25 machines with varying amounts of data in memory

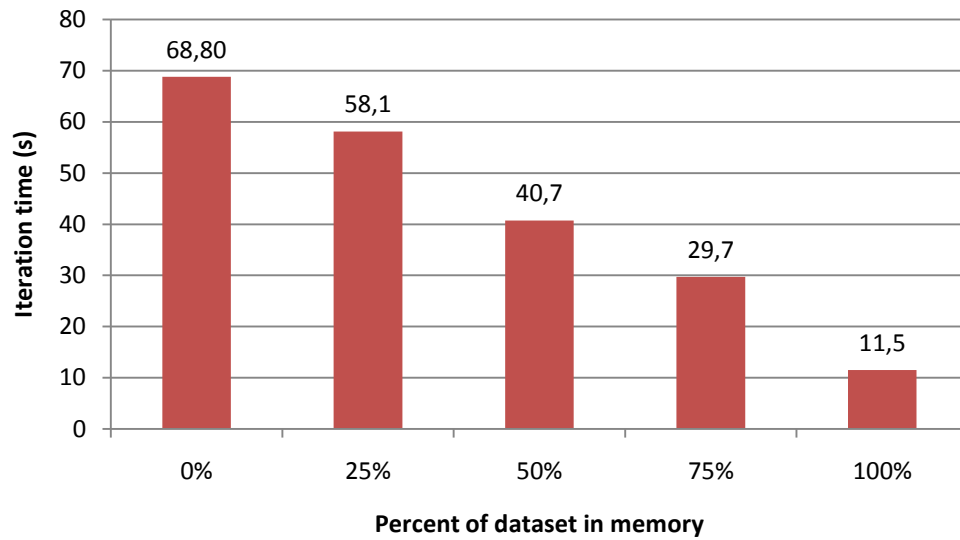


Abbildung 7: Verarbeitungsgeschwindigkeit bei unterschiedlicher Datenmenge im Speicher

Performance Page Rank

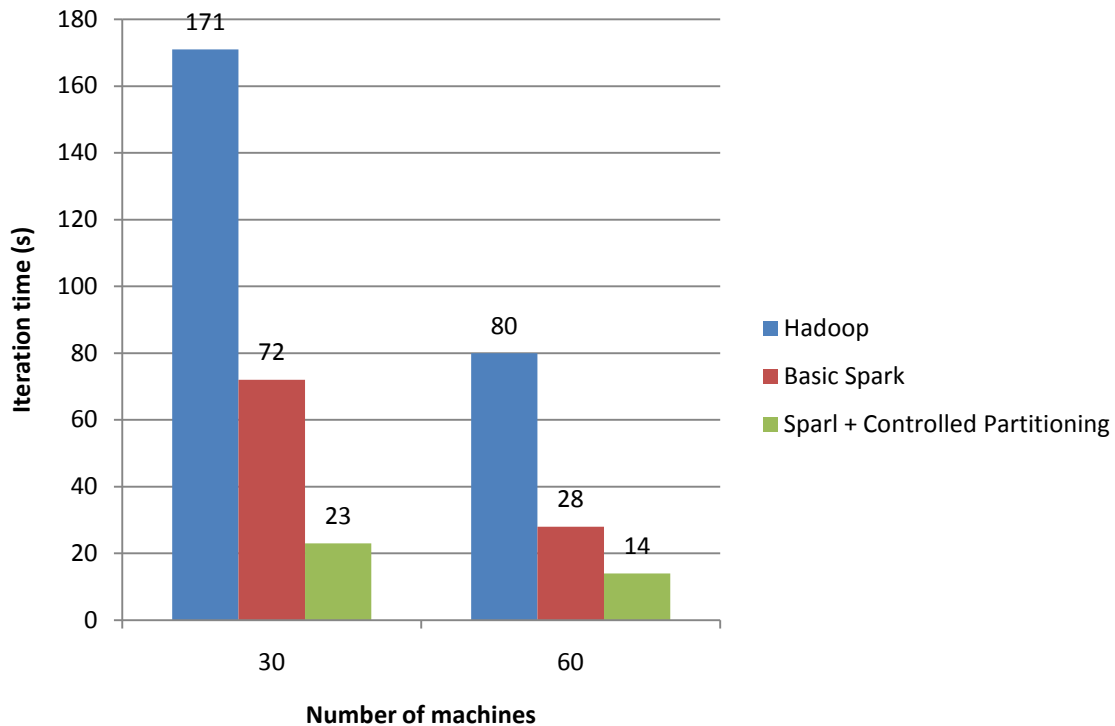


Abbildung 8: Verarbeitungsgeschwindigkeit bei unterschiedlicher Anzahl an Maschinen im Vergleich

Ausblick

Der Abschnitt Ausblick wird sich im ersten Teil mit dem ebenfalls von der UC Berkeley entwickelten Plattform Shark beschäftigen. Dieses ist gleichzusetzen mit der von Apache angebotenen Hive Architektur und basiert auf Resilient Distributed Datasets. Der Besucher erhält eine Architekturübersicht und Vorteile zu der von Apache angebotenen Plattform. Der zweite Teil ist für Quellen und weiterführende Links vorbehalten.

Kontaktadresse:

Dirk Reinemann
Oracle Deutschland B.V. & Co. KG
Schiffbauergasse 14
D-14467 Potsdam

Telefon: +49 (0) 331-200 7454
Fax:
E-Mail: dirk.reinemann@oracle.com
Internet: