Big Data

Isabella Aspodinger, Alexander Pilan

Paris Lodron Universität Salzburg

22. November, 2019

Inhalt

Allgemein

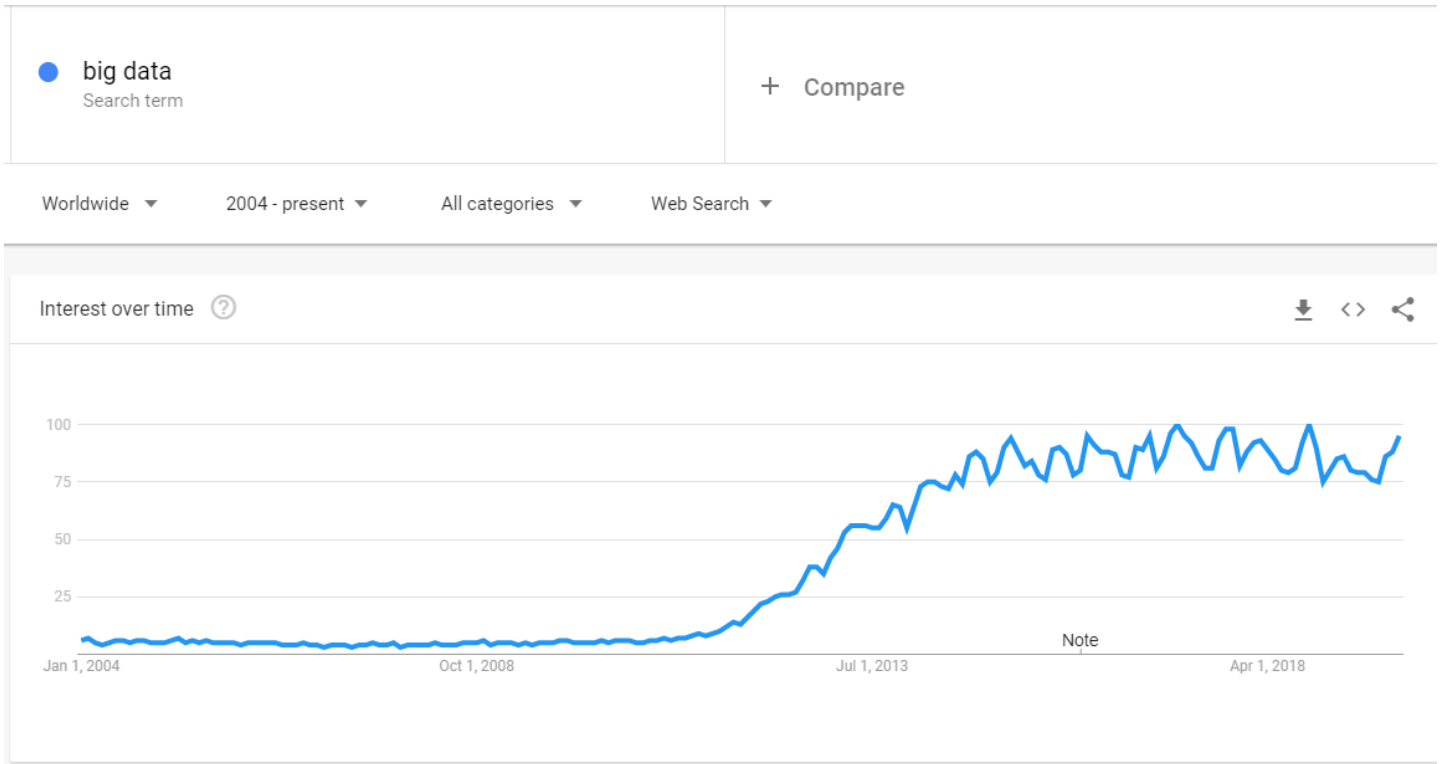


Figure: https://trends.google.com/trends/

Definition

- Volume (Datenvolumen)
- Velocity (Geschwindigkeit der Datenverarbeitung und Veränderungsdynamik)
- Variety (Vielfalt der Datenstrukturen und -klassen)
- Veracity (Echtheit der Daten)
- Value (unternehmerischer Mehrwert)
- Validity (Datenqualität)

Unterschied

Traditionelle Analytik

- Schrittweise Analyse von kleinen Datenmengen
- Daten werden angesammelt, bearbeitet, gespeichert und erst dann analysiert

Big Data Analytik

- Bearbeitung der ganzen Datenmenge
- Analyse und Bearbeitung werden je nach Eingang durchgeführt

Datenherkunft

1. Aufzeichnungen verschiedenster Überwachungssysteme.
2. die Nutzung von Kunden- oder Bank- bzw. Bezahlkarten
3. die Nutzung eines Smartphones
4. Social-Media
5. Kraftfahrzeuge
6. vernetzte Technik in Häusern
7. von Behörden und Unternehmen erhobene und gesammelte Daten.

Wachstum von Daten

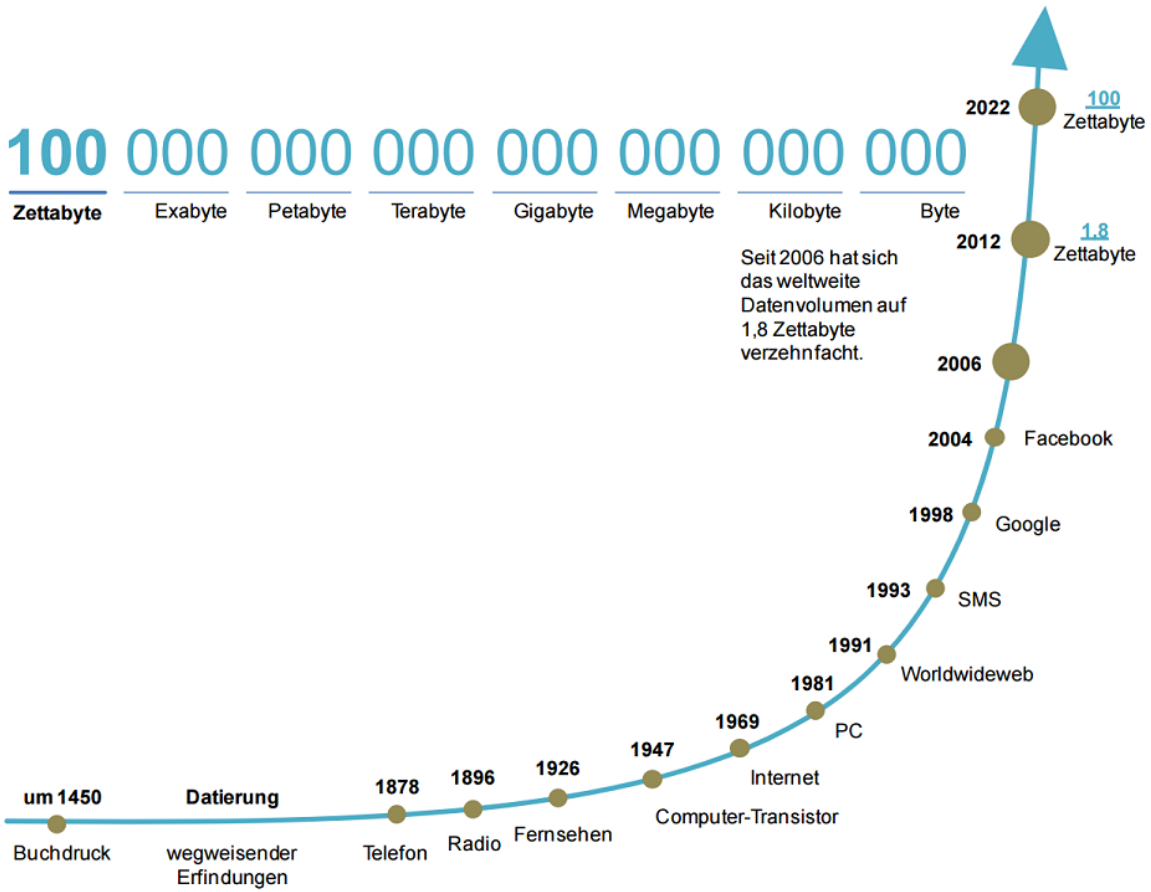


Figure:
https://bigdatablog.actgruppe.de/geschaeftliche-herausforderung-von-big-data/

Anwendung

1. Wahlen
2. Social Scoring
3. Bildungswesen
4. Wirtschaft
5. Staat
6. Gesundheit
7. Umwelt

Analysemethoden

- Repräsentative Stichprobe
- Data Mining
- Predictive Analytics
- Crowdsourcing
- Datenfusion und -integration
- Machine Learning
- Neuronale Netze

Entwicklungen

- NOSQL (Not Only SQL)
- JSON
- Map Reduce
- Hadoop
- Spark
- R

NoSQL

- Datenbanken die meist
  - nicht-relational
  - verteilt
  - open source und
  - horizontal skalierbar sind
- ursprünglicher Fokus: moderne “web-scale” Datenbanken
- Entwicklung seit ca. 2009
- Charakteristika:
  - schema-frei, Datenreplikation, einfache API
  - eventually consistent / BASE (statt ACID)

NoSQL

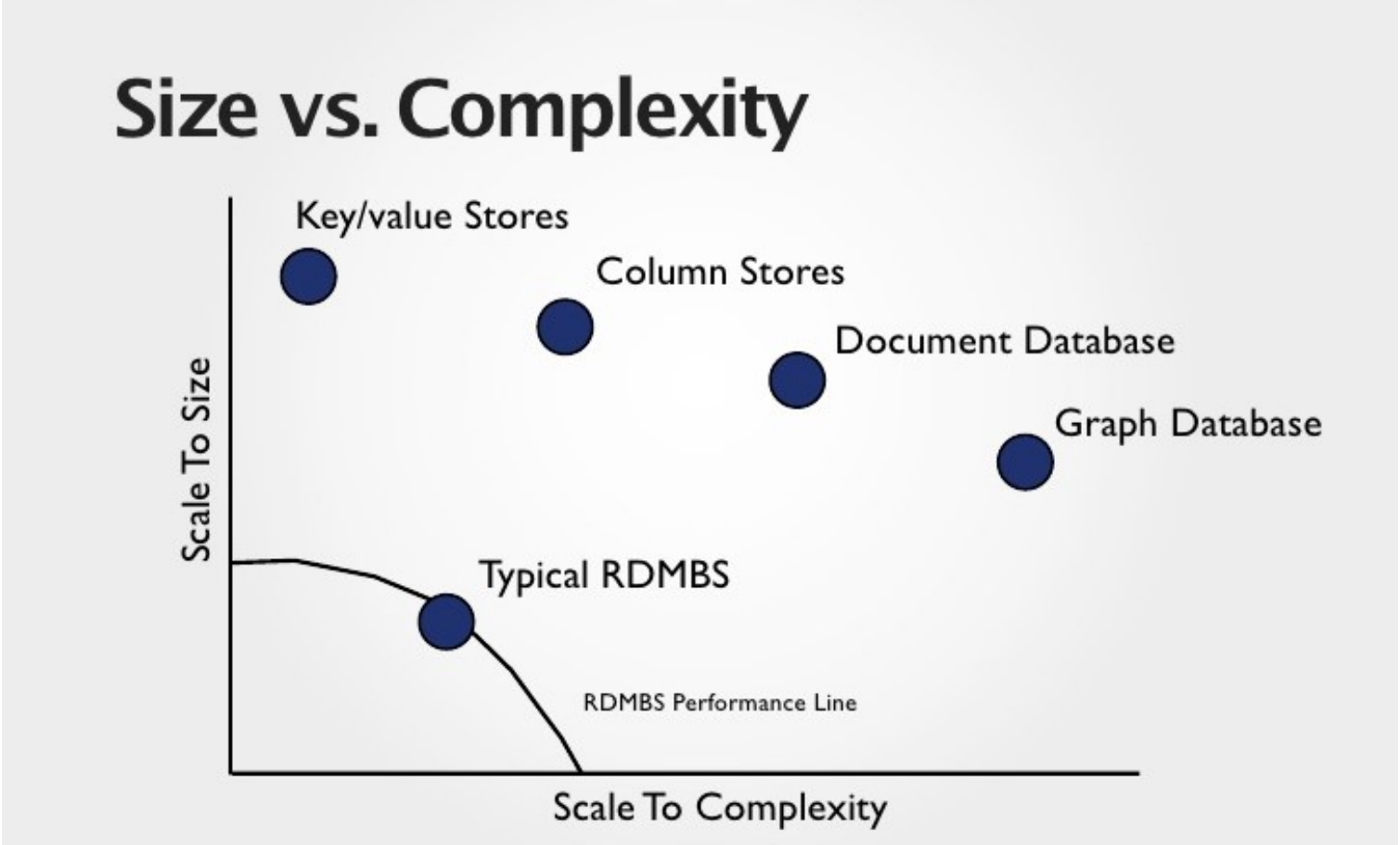


Figure: https://www.slideshare.net/timjuravich/mysql-nosql-from-a-php-perspective

JavaScript Object Notation

JSON ist ein kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen.

```
{
  "Herausgeber": "Xema",
  "Nummer": "1234-5678-9012-3456",
  "Deckung": 2e+6,
  "Waehrung": "EURO",
  "Inhaber": {
    "Name": "Mustermann",
    "Vorname": "Max",
    "maennlich": true,
    "Hobbys": ["Reiten", "Golfen", "Lesen"],
    "Alter": 42,
    "Kinder": [],
    "Partner": null
  }}
```

Map Reduce

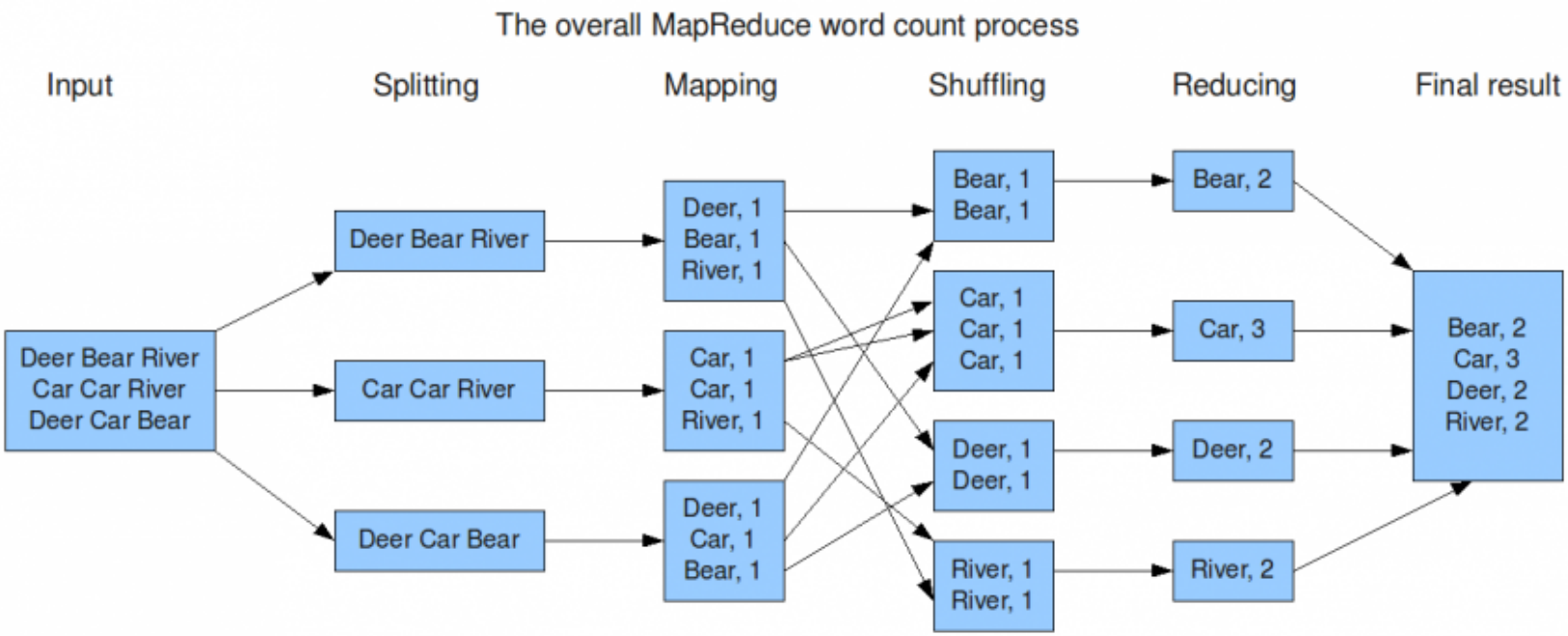


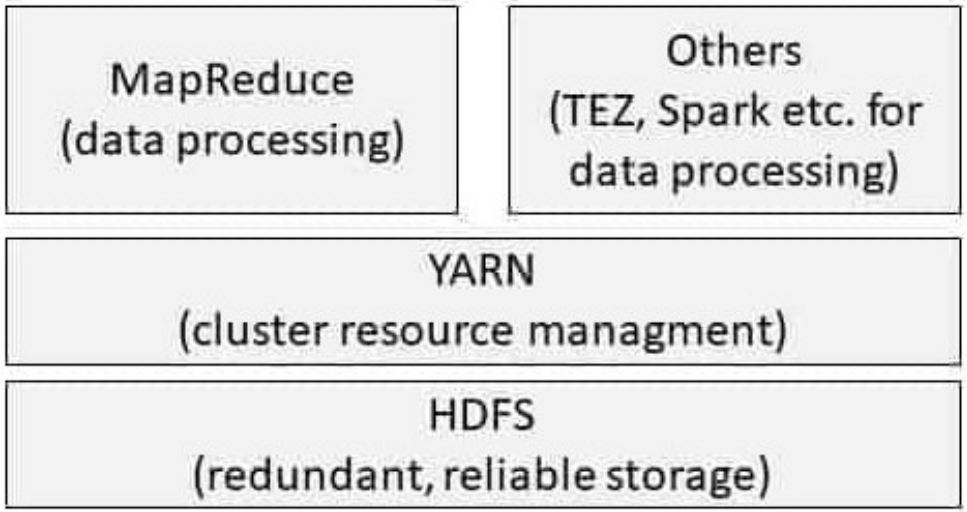
Figure: http://beyondthegeek.com/2016/07/13/mapper-reducer/

Map Reduce

Wortzahl

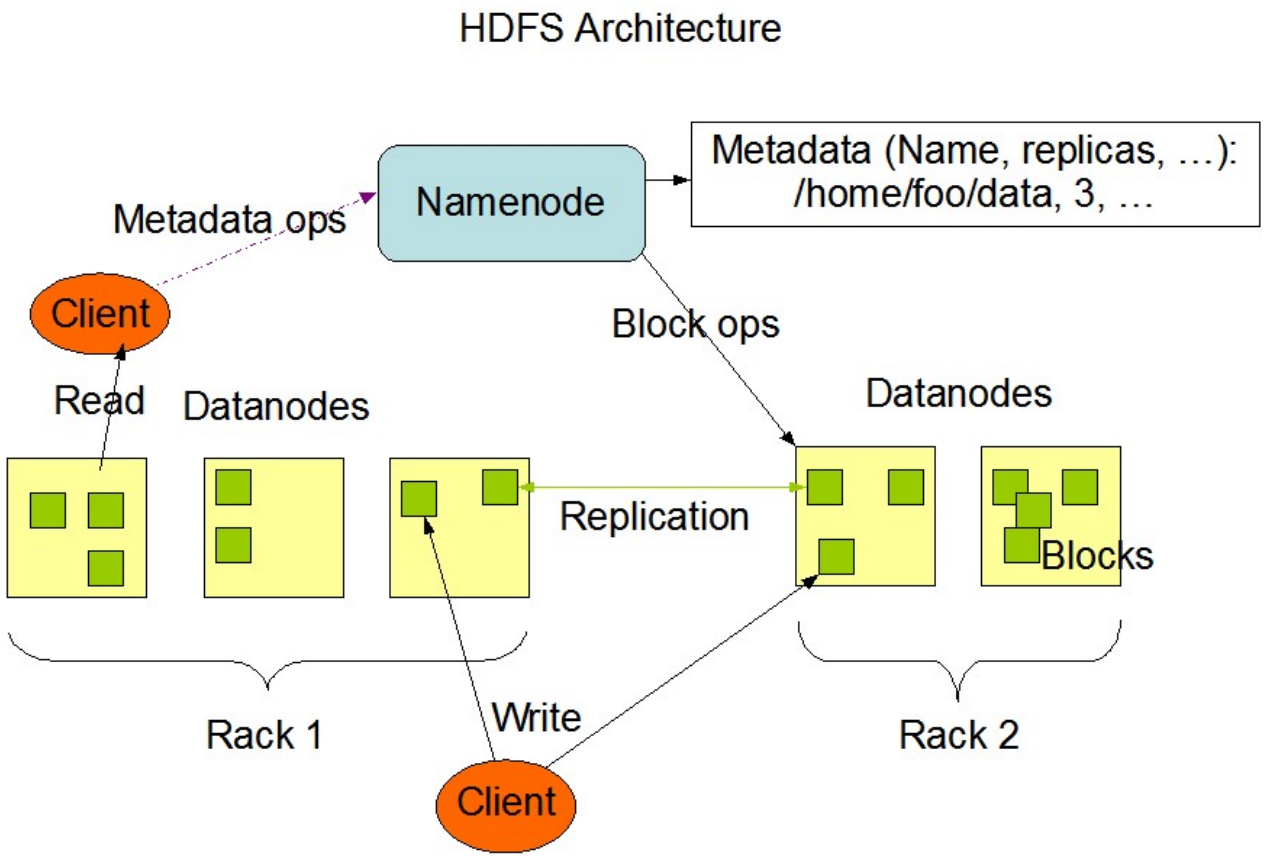
```
1 function map(String name, String documentPart):
2         for each word w in documentPart:
3                 emit (w, 1)
4
5 function reduce(String word, List<Int> partialCounts):
6         sum = 0
7         for each pc in partialCounts:
8                 sum += pc
9                 emit (word, sum)
```

Hadoop



- HDFS (Hadoop Distributed File System)
- YARN (Yet Another Resource Negotiator)
- Map Reduce

HDFS


HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Client
Rack 2

R

Paradigmen:

- funktional
- dynamisch
- objektorientiert

R Beispiel Code

```r
Gewicht <- c(60, 72, 57, 90, 95, 72)
Groesse <- c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
BMI <- Gewicht/ Groesse^2
sum(Gewicht)
length(Gewicht)
sum(Gewicht)/length(Gewicht)
table(Gewicht)
```

Spark

- einheitliches In- Memory System
- zur Verarbeitung von enormen Datenmengen geeignet
- Framework für Clustercomputing
- Open Source
- Konzept der Resilient Distributed Datasets (RDD)

Spark

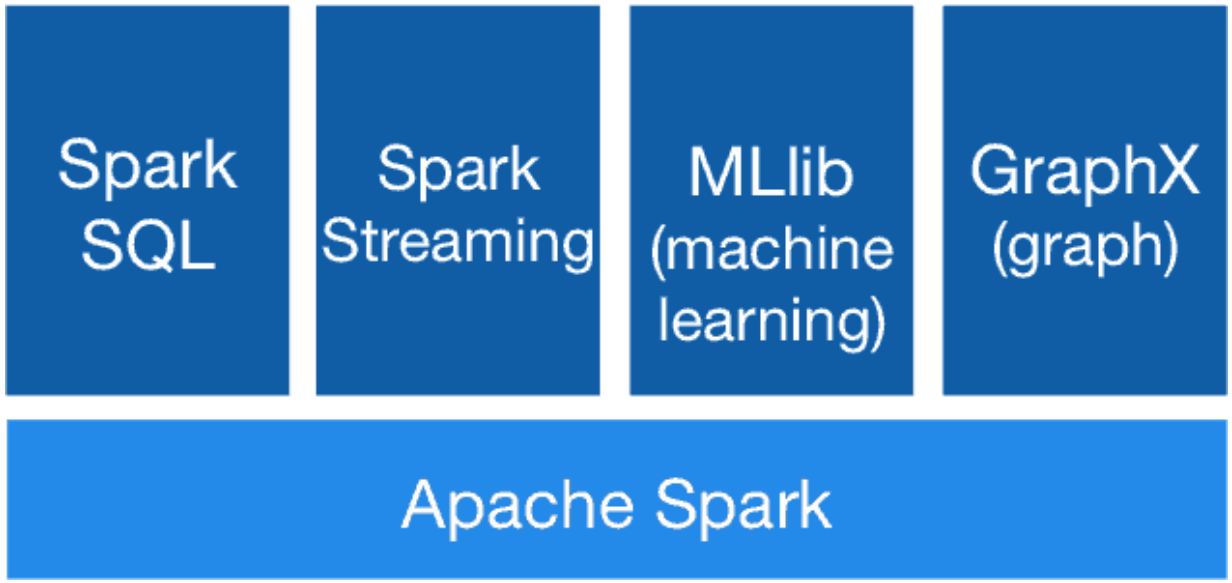


Figure: http://archive.ibmsystemsmag.com/mainframe/business-strategy/bi-and-analytics/dynamic-spark/

Zusammenfassung

Schöne neue Welt?

Danke für Ihre Aufmerksamkeit!