Introduction to Apache Spark

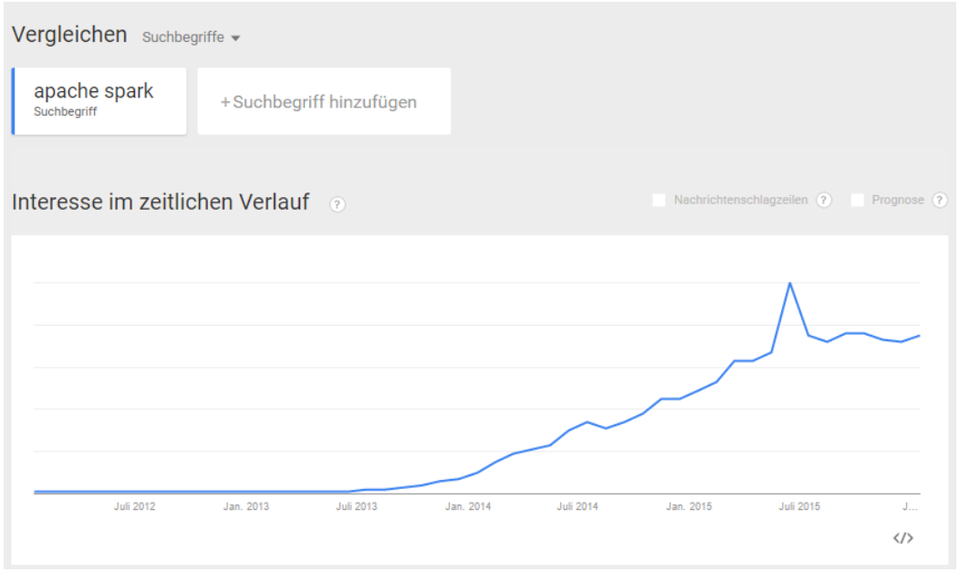
dsquare.de
Salzburger Straße 27
83073 Stephanskirchen

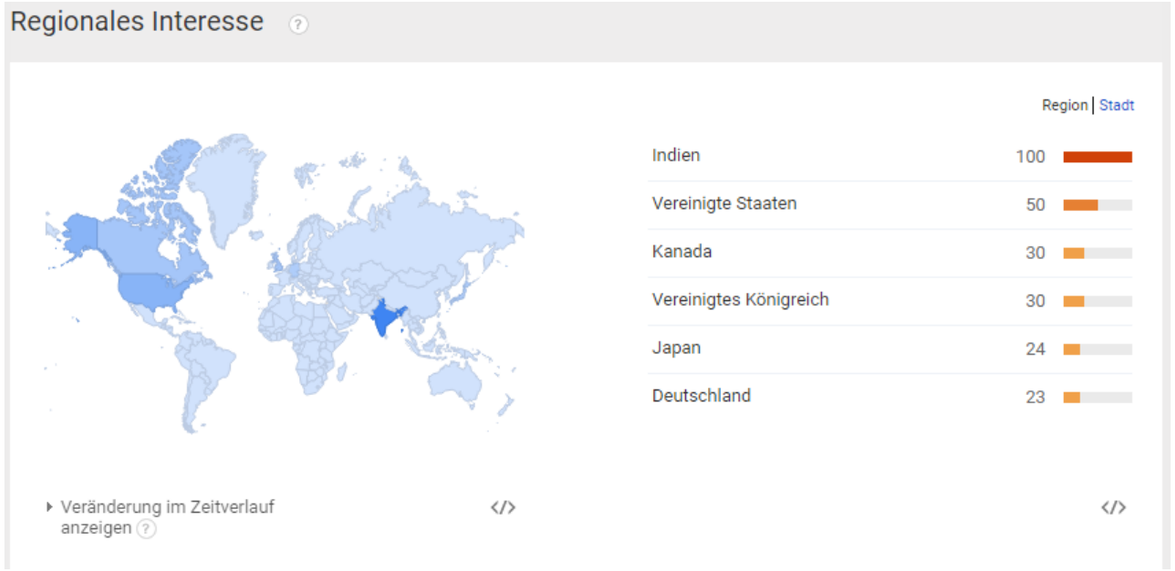
Tel.: 08031-234 1140
Mobil: 0172-1484 731
Email: info@dsquare.de
www.dsquare.de

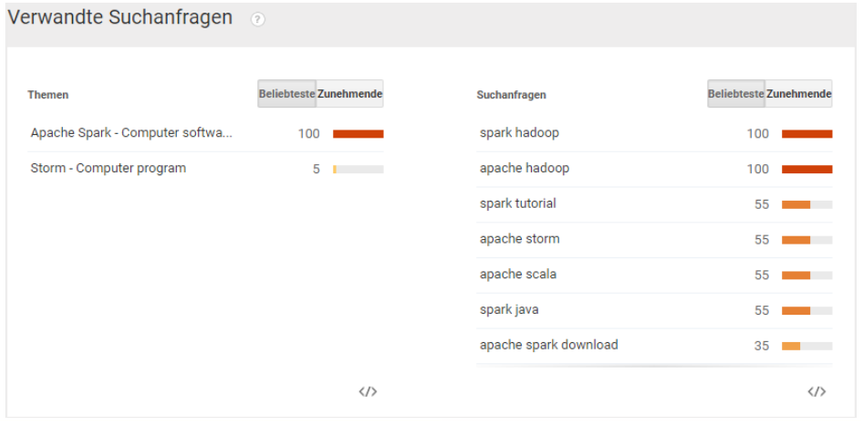
dsquare.de
DATA IN MIND

Rosenheim, 2. August 2016

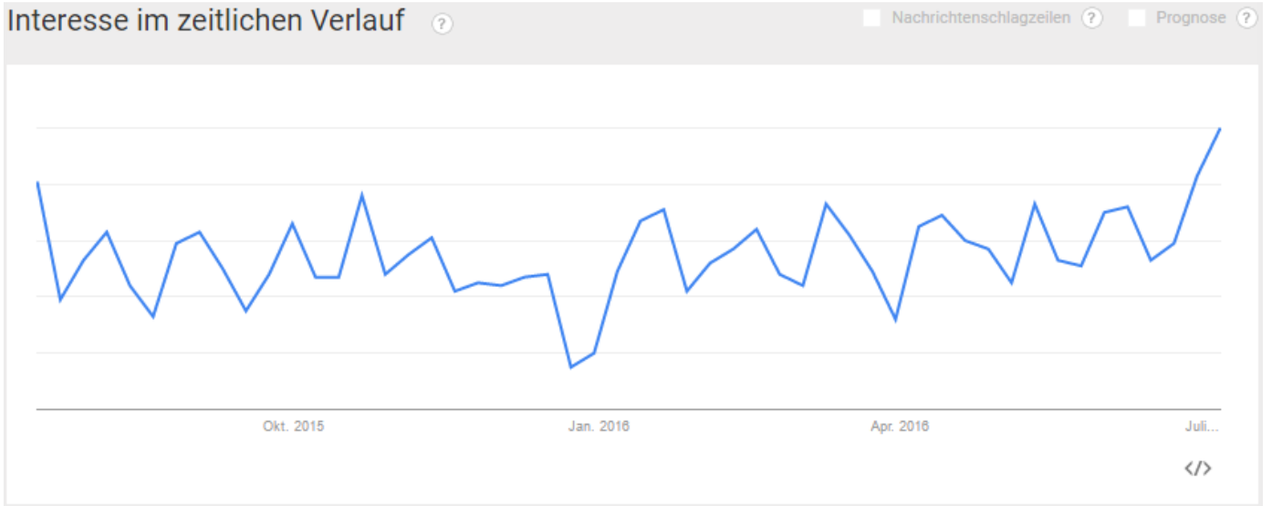
Apache Spark according to Google Trends





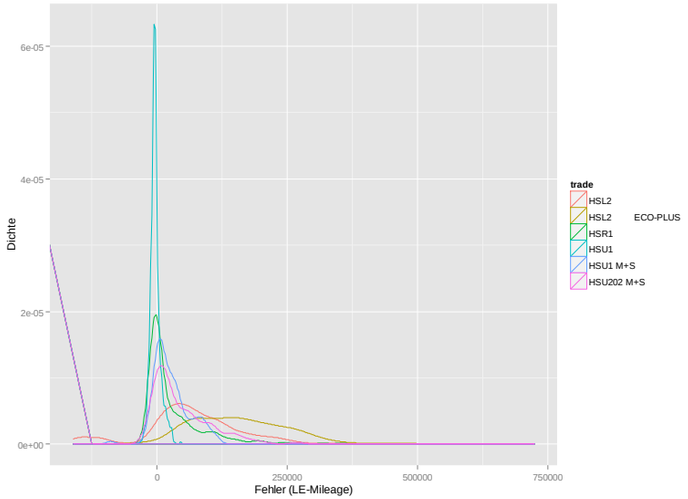


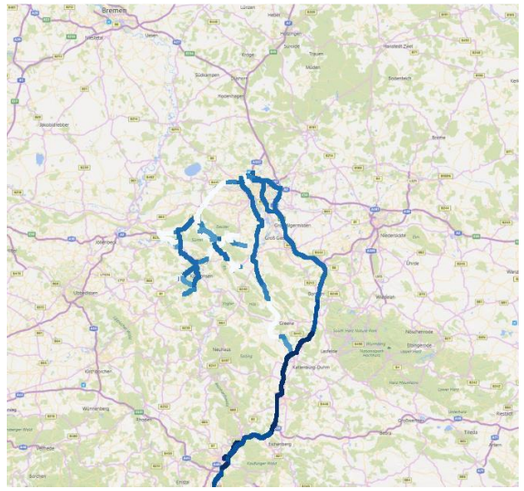
TDWI Conference sparked interest in Spark



Problem

- Huge amount of GPS data
- Huge amount of sensorial data
- Need to process GPS very fast
- Kernel density estimation on spatial data
- Panel data on Huge amount of object with multiple measurements





What is Apache Spark?

- A cluster-based computing engine
- Developed since 2012
- Developed by students at UC Berkley
- APIs for
  - Python
  - Java
  - R
  - Scala
- Supports SQL, ML, Streaming Data, Graph processing
- Faster than Hadoops Map-Reduce

Timeline

Since late 1990s
APPLY Functions
In-memory
Single Process
Single Core

Not Scalable

Since 2007
Map-Reduce
Parallel Computing
Distributed File System

Linear Scalability

Since 2009
Directed acyclic graph
Parallel Computing
Distributed File System

Linear Scalability

Map-reduce vs. Spark

Map-reduce

Directed acyclic graph
- No writeback to HDFS necessary
- Data passed to next processing step

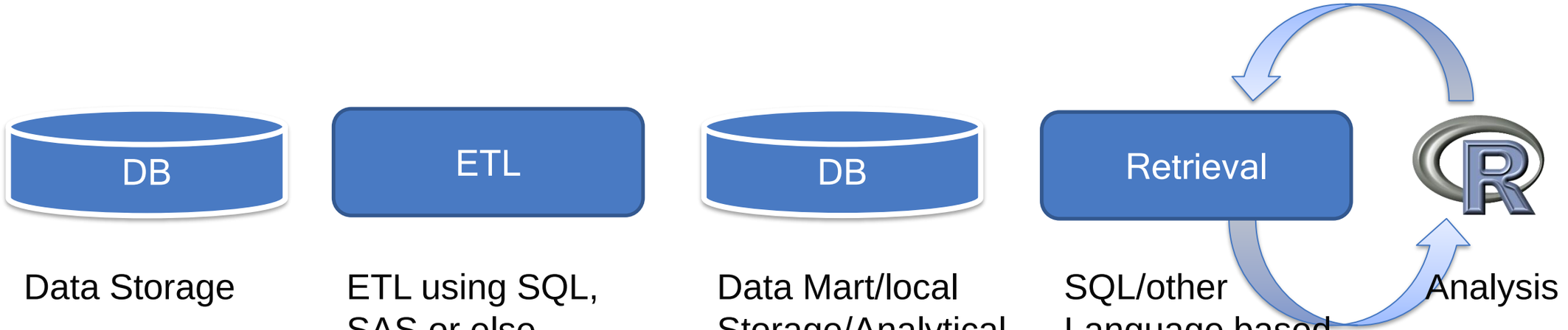
Developer focused
- Transformations available
- Many APIs

In-Memory processing
- RDD materialized in memory across cluster
- No need to reload from disc

Spark is well suited for the needs of Data Scientists

Iterative application of algorithms
Multiple passes over data sets
Reactive applications

Spark can unify an analytical environment

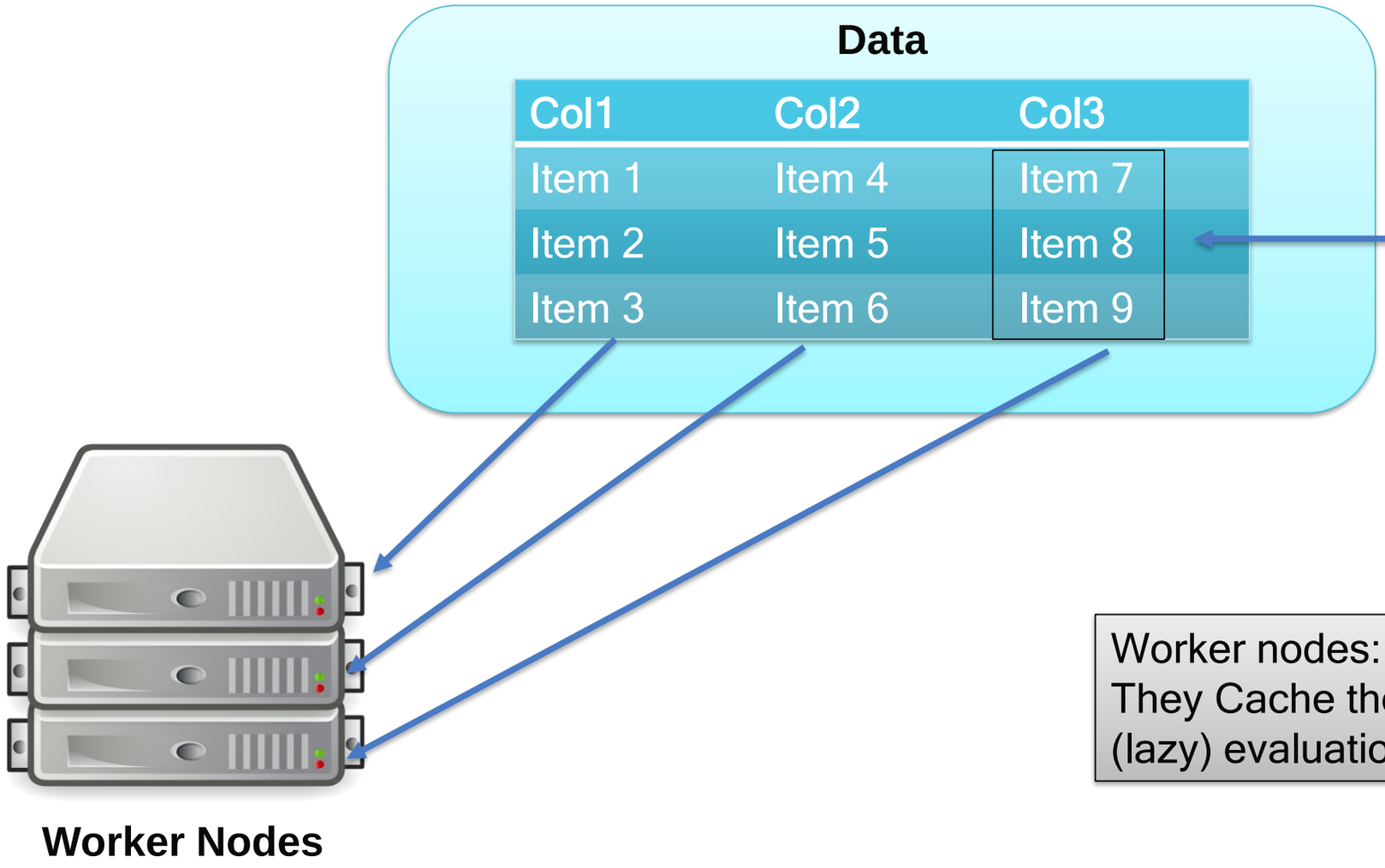


SparkR

Spark

REPL = Read Evaluate Print Loop (back)

RDD


Data
Col1
Col2
Col3
Item 1
Item 4
Item 7
Item 2
Item 5
Item 8
Item 3
Item 6
Item 9
This could be an RDD = Resiliant Distributed Dataset
Worker nodes:
They Cache the data and do the (lazy) evaluation.
Worker Nodes

Preliminaries using Spark in R-Studio

```
.libPaths(c(.libPaths(), '/opt/spark-1.6.1-bin-hadoop2.6/R/lib'))Sys.setenv(SPARK_HOME =
'/opt/spark-1.6.1-bin-hadoop2.6')
Sys.setenv(PATH = paste(Sys.getenv(c('PATH')), '/opt/spark-1.6.1-bin-hadoop2.6/bin', sep =
':'))
library(SparkR)

d.csv <- "com.databricks:spark-csv_2.11:1.4.0„
d.pg <- "org.postgresql:postgresql:9.4.1208"

sc <- sparkR.init(sparkPackages=c(d.csv))
sqlContext <- sparkRSQL.init(sc)
```

Get data from Spark

```
s.df <- read.df(sqlContext,
        source = "com.databricks.spark.csv",
        path = "/var/data/server-sample.log",
        delimiter = " ", header = "false")

cache(s.df) # Bring Spark data.frame to R
registerTempTable(s.df, "logs")

rc <- sql(sqlContext, "SELECT C0 AS ip, COUNT(*) AS n FROM logs GROUP BY C0 ORDER BY
COUNT(*) DESC")
```

Analysing Sensorial Data

GPS
Pressure
Temperature
Acceleration
...

Hadoop Infrastructure

Hadoop
HDFS
Storm
Hive
Spark

Spark

R

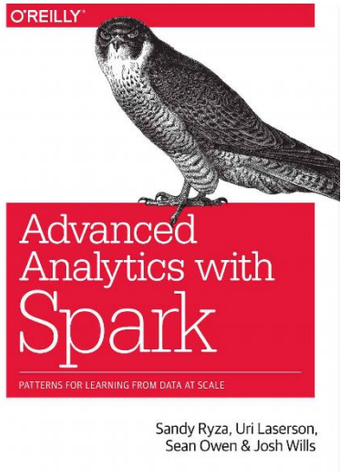
Developing predictive models

R&D Dashboards

Other Datasources (Weather, Height Data, GIS Data

Further Sources

https://spark.apache.org/docs/latest/api/R/index.html
https://spark.apache.org/docs/latest/sparkr.html



dsquare.de

DATA IN MIND

Für Fragen stehen wir Ihnen gerne zur Verfügung!