Scaling SparkR in Production. Lessons from the Field.

Heiko Korndorf
Wireframe, CEO & Founder

SPARK SUMMIT EUROPE 2016

Who Am I

Heiko Korndorf

- Founder Wireframe
- MS in Computer Science
- Application Areas: ERP, CRM, BI, EAI
- Helping companies in
  - Manufacturing
  - Telecommunications
  - Financial Services
  - Utilities
  - Oil & Gas
  - Professional Services

Content

Classify this talk ....

- Data Science: Scaling your R application with SparkR
- Data Engineering: How to bring Data Science applications into your production pipelines, i.e. adding R to your toolset.
- Management: Integrating Data Science and Data Engineering with SparkR

Agenda

- R and Real-world Projects I + II
- SparkR Architecture 1.x/2.x
- Approach with Spark 1.5/1.6
  - Parallelization via YARN
  - Dynamic R Deployment, incl. dependencies/packages
  - R-Graphics: headless environment, concurrency
- Approach with Spark 2.0
  - Parallelization via SparkR
  - Use Spark APIs: SQL, Mllib
  - On-Prem vs Cloud (Elasticity/decouple storage and compute)
- Integrating Data Science and Data Engineering
- A Broader Look at the Ecosystem
- Outlook and Next Steps

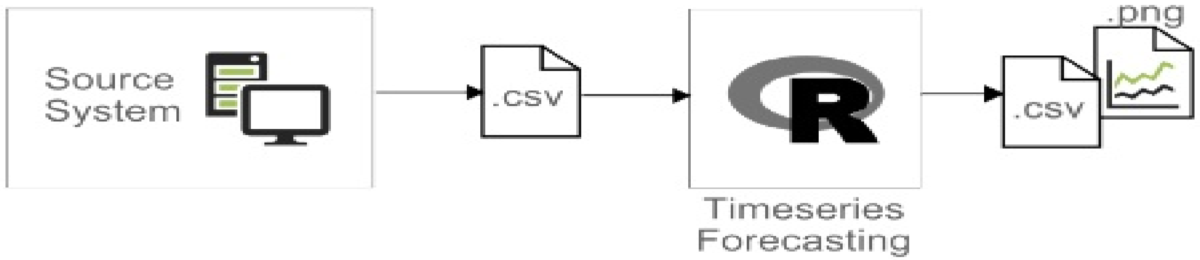
Data Science with R

- Very popular language
- Designed by statisticians
- Large community
- > 10.000 packages
    - plus: integrated package management

- But: Limited as Single-Node platform
    - Data has to fit in memory
    - Limited concurrency for processing

SparkR Projects

Project I
- Compute-Intensive
- On-Prem / MS Azure
- Monthly Runs

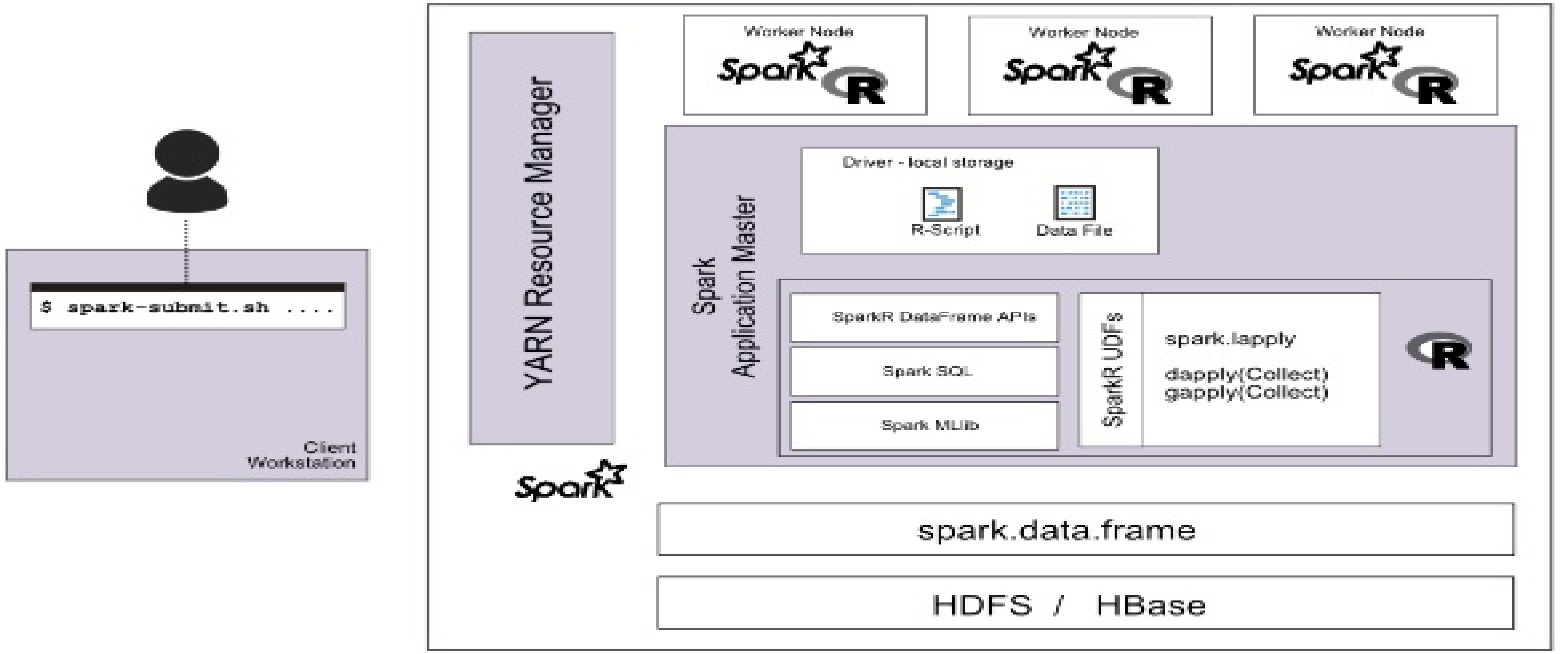


Project II
- Data Integration
- On-Prem / Amazon AWS
- Monthly/Daily Runs

SparkR from an R Perspective

- Import SparkR-package and initialize SparkContext and SQLContext
- Access Data stored in Hadoop, HBase, Cassandra, etc.
- Use Spark Libraries
- Parallelize R

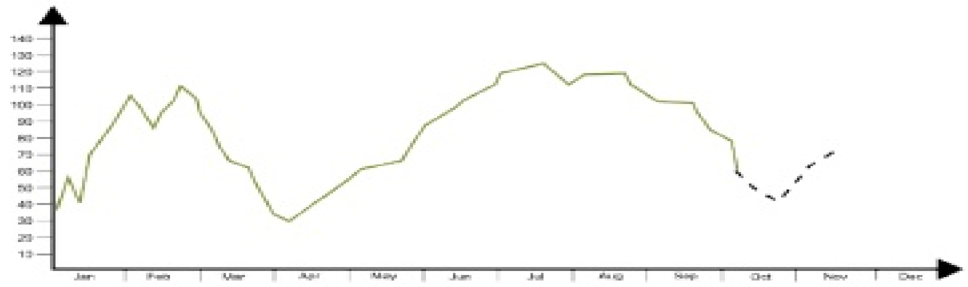
SparkR Architecture 1.x/2.x



Time Series Forecasting

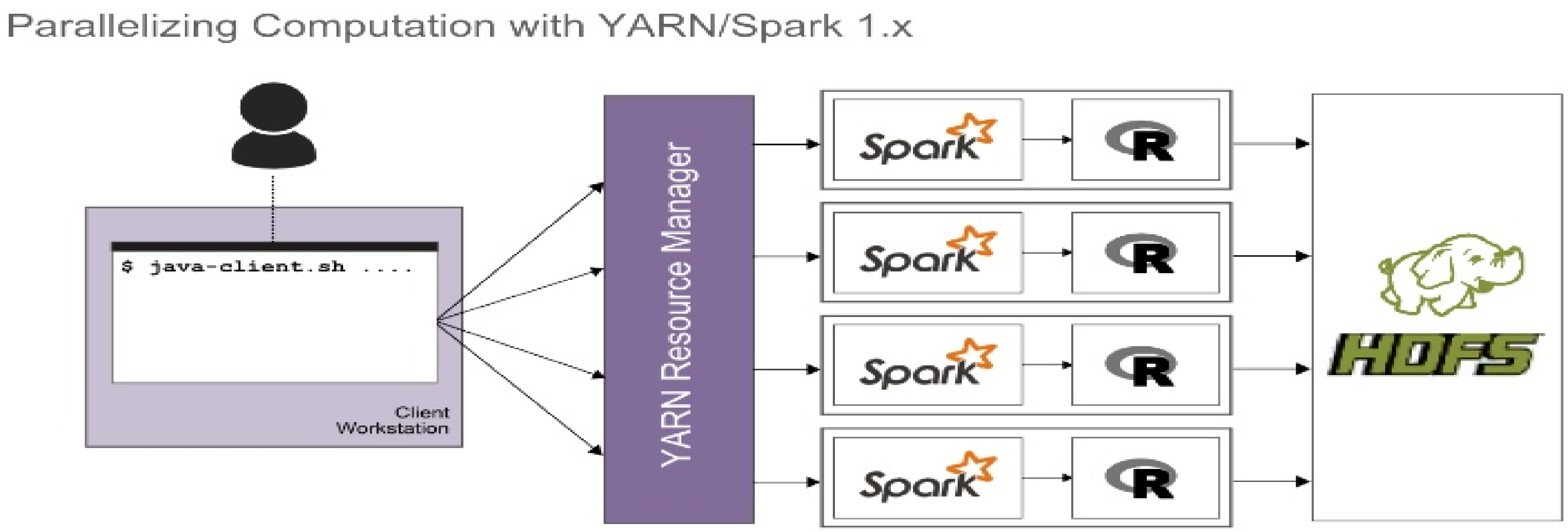
- Time Series: a series of data points indexed in time order
- Methods:
  - Exponential Smoothing
  - Neural Networks
  - ARIMA:

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) (1 - L)^d X_t = \delta + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

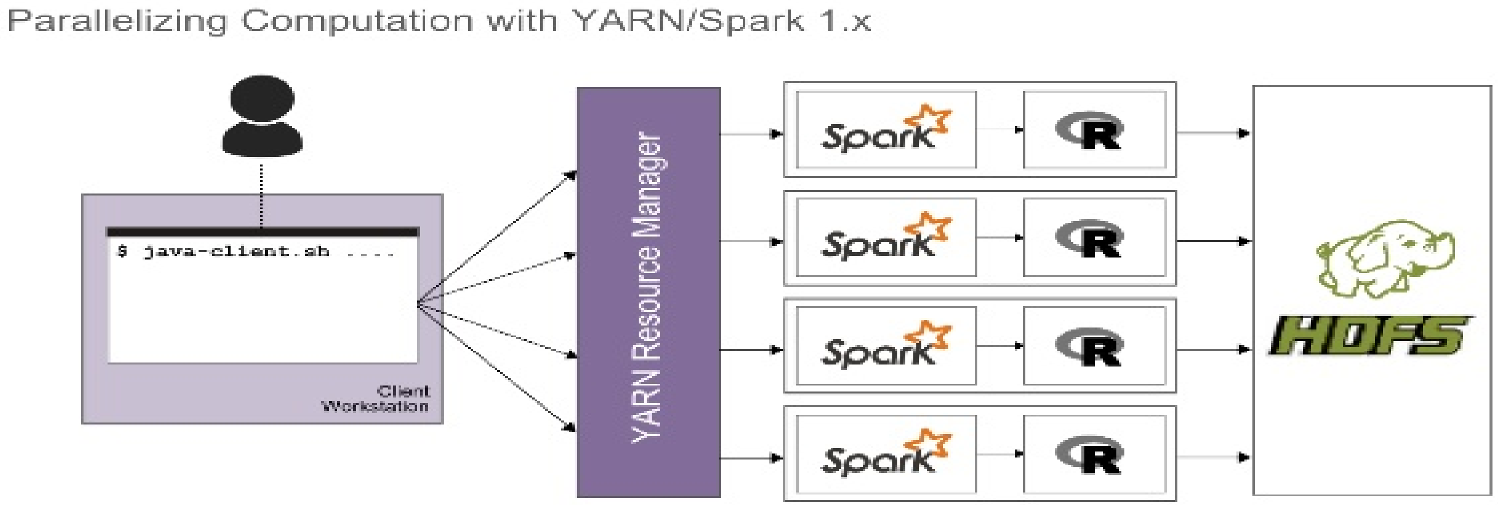


- ARIMA(p,d,q)
  - AR: p = order of the autoregressive part
  - I: d = degree of first differencing involved
  - MA: q = order of the moving average part

Parallelization with SparkR 1.x



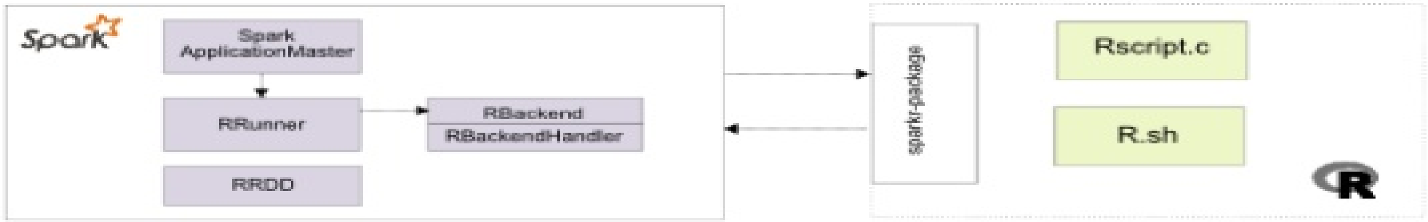
Parallelization with SparkR 1.x



- Sequential computation: > 20 hrs.
- Single-Server, parallelized: > 4.5 hrs
- SparkR 1.6.2, 25 nodes, 4 cores: ca. 12 mins.

Dynamic R Deployment

- Challenge: R not installed on cluster

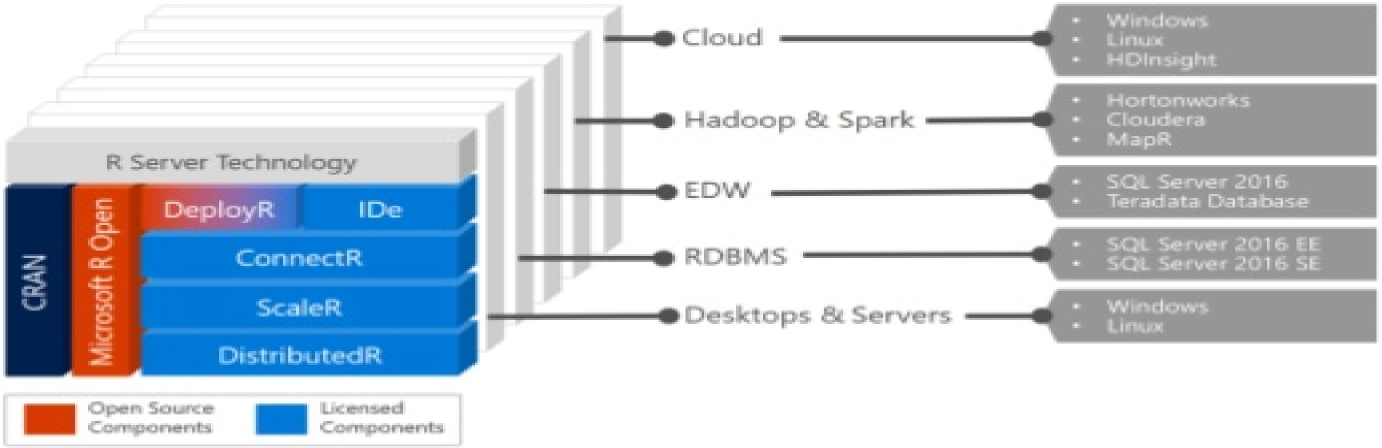


- R's installation location is hard-coded in R
- Modify some Spark and R source files and build Spark and R from source
- Include dependant packages in R
- Submit Spark job with R as YARN dependency

Microsoft R Server HDInsight

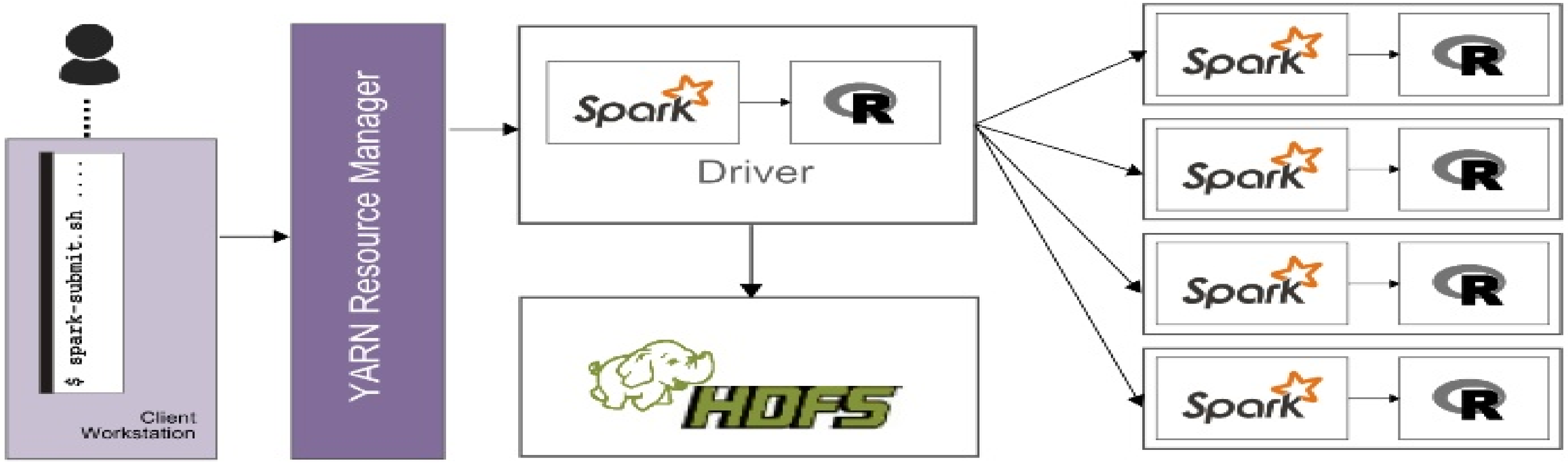
- Microsoft R Server for HDInsight integrates Spark and R
- Based on Revolution Analytics

- UDFs via rxExec()



Parallelization with SparkR 2.0

Parallelizing Computation with YARN/SparkR 2.x



Parallelization with SparkR 2.0

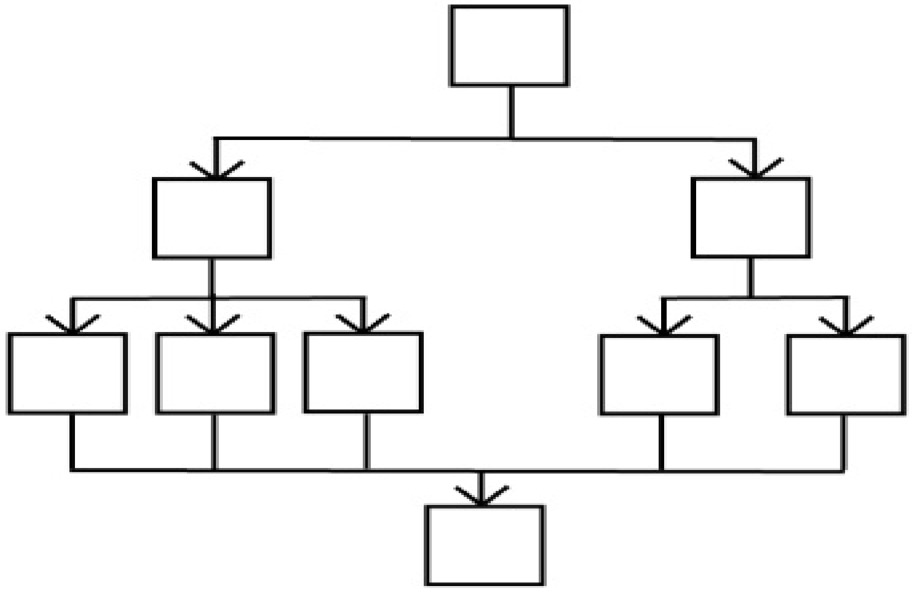
Support for User-Defined Functions

- dapply / dapplyCollect
  - input: DataFrame, func [, Schema]
  - output: DataFrame
- gapply / gapplyCollect
  - input: DataFrame¦GroupedData, groupBy, func [, Schema]
  - output: DataFrame
- spark.lapply
  - input: parameters, func
    - Access to data/HDFS
  - output: List

Advanced Parallelization

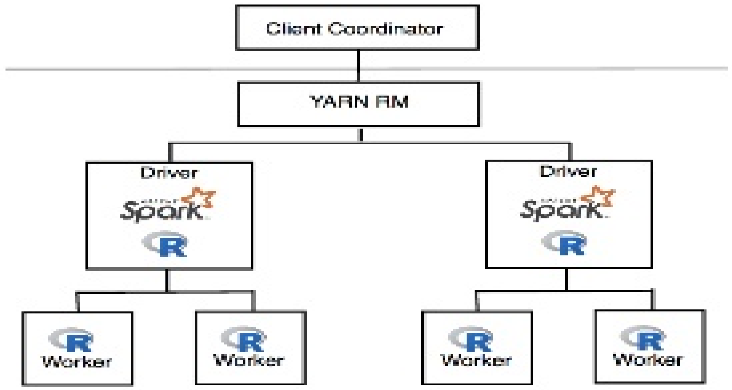
Not only Data-Parallel vs Model-Parallel ....
More generic than Map-Reduce

- Coordinating Computation
- Managing Data I/O

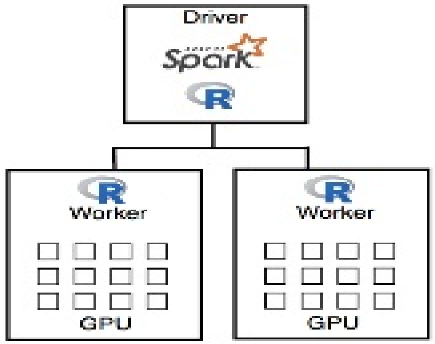


2-Level Parallelization

(1) Submit multiple jobs to your cluster:
- Cluster Manager (YARN, Spark, Mesos)
- Spark Job: Driver and Executors



(2) Use GPGPU
- Spark Job: Driver and Executor
- Let Executor use GPGPU



(3) Combine 1 and 2

Cultural Integration

Data Science

- Exploratory
- Loosely structured
- Higher-level abstractions
- Less concerned with performance and efficiency

Data Engineering

- Precision
- Robustness & Repeatability
- Focus on efficiency and performance

Spark & R – A Dynamic Ecosystem

Other Interesting Projects in the Spark/R Ecosystem:

- SystemML
- Renjin
- sparklyr
- RHIPE (R on Hadoop)

- Spark Time Series Libraries (spark-timeseries, huohua/Flint)

Outlook & Next Steps

- Organizational: Further Integrate Data Engineering & Data Science
  - Source Code Control & Versioning
  - Continuous Build
  - Test Management

- Technical: New Approaches
  - Simplify/Unify Data Pipelines (SparkSQL)
    - Mix Spark/Scala and R!
    - R Graphics: raster images to data frame
  - Performance Improvement: use MLlib
  - Performance Improvement: move calculation to GPU

Questions?

THANK YOU.

Heiko Korndorf

heiko.korndorf@wireframe.li

SPARK SUMMIT
EUROPE 2016