KNIME Big Data Workshop

Tobias Kötter and Björn Lohrmann

KNIME
Variety, Velocity, Volume

• Variety:
  – Integrating heterogeneous data...
  – ... and tools

• Velocity:
  – Real time scoring of millions of records/sec
  – Continuous data streams
  – Distributed computation

• Volume:
  – From small files...
  – ...to distributed data repositories
  – Moving computation to the data
Variety
The KNIME Analytics Platform: Open for Every Data, Tool, and User

KNIME Analytics Platform

Data Scientist

Business Analyst

External Data Connectors

Native Data Access, Analysis, Visualization, and Reporting

External and Legacy Tools

Distributed / Cloud Execution

mongoDB

Hive

Impala

Hortonworks

Cloudera

TIBCO Spotfire

ACTUATE BiRT

R

Python

SPSS

Open for Innovation

© 2017 KNIME.com AG. All Rights Reserved.
# Data Integration

<table>
<thead>
<tr>
<th>Sepal length</th>
<th>Sepal width</th>
<th>Petal length</th>
<th>Petal width</th>
<th>Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>3.5</td>
<td>1.4</td>
<td>0.2</td>
<td><em>I. setosa</em></td>
</tr>
<tr>
<td>4.9</td>
<td>3.0</td>
<td>1.4</td>
<td>0.2</td>
<td><em>I. setosa</em></td>
</tr>
<tr>
<td>4.7</td>
<td>3.2</td>
<td>1.3</td>
<td>0.2</td>
<td><em>I. setosa</em></td>
</tr>
<tr>
<td>4.6</td>
<td>3.1</td>
<td>1.5</td>
<td>0.2</td>
<td><em>I. setosa</em></td>
</tr>
<tr>
<td>5.0</td>
<td>3.6</td>
<td>1.4</td>
<td>0.2</td>
<td><em>I. setosa</em></td>
</tr>
<tr>
<td>5.4</td>
<td>3.9</td>
<td>1.7</td>
<td>0.4</td>
<td><em>I. setosa</em></td>
</tr>
<tr>
<td>4.6</td>
<td>3.4</td>
<td>1.4</td>
<td>0.3</td>
<td><em>I. setosa</em></td>
</tr>
<tr>
<td>5.0</td>
<td>3.4</td>
<td>1.5</td>
<td>0.2</td>
<td><em>I. setosa</em></td>
</tr>
</tbody>
</table>
Integrating R and Python
Modular Integrations

R Snippets
- R Snippet
- R View (Table)
- R Source (Table)

R Workspace Manipulation
- R Source (Workspace)
- R To R
- R View (Workspace)
- R Model Reader
- Table to R
- Add Table To R
- R to Table
- R Model Writer
- R Learner
- R Predictor
- R To PMML

Python Snippets
- Python Script
- Python View
- Python Edit Variable
- Python Script (2:1)
- Python Source

Python Learner
- Python Script (DB)
- Python Object Reader
- Python Predictor
- Python Script (Hive)
- Python Object Writer
Other Programming/Scripting Integrations

Java Integrations
- Java Snippet
- Java Snippet Row Filter
- Java Snippet Row Splitter

JavaScript Integrations
- Generic JavaScript View
- JavaScript Table View
- JavaScript Scatter Plot
- JavaScript Lift Chart
- JavaScript Conditional Box Plot
- JavaScript ROC Curve
- JavaScript Line Plot
- JavaScript Box Plot
- JavaScript Bar Chart
- JavaScript Pie/Donut Chart

Misc Integrations
- External Tool
- External SSH Tool
- Generic Web Service Client
- SAS7BDAT Reader
Velocity
Velocity

• High Demand Scoring/Prediction:
  – High Performance Scoring using generic Workflows
  – High Performance Scoring of Predictive Models

• Continuous Data Streams
  – Streaming in KNIME

• Distributed Computation
  – KNIME Cluster Executor
High Performance Scoring via Workflows

- Record (or small batch) based processing
- Exposed as RESTful web service
High Performance Scoring using Models

• KNIME PMML Scoring via compiled PMML
• Deployed on KNIME Server
• Exposed as RESTful web service

• Partnership with Zementis
  – ADAPA Real Time Scoring
  – UPPI Big Data Scoring Engine
Velocity

• High Demand Scoring/Prediction:
  – High Performance Scoring using generic Workflows
  – High Performance Scoring of Predictive Models

• Continuous Data Streams
  – Streaming in KNIME

• Distributed Computation
  – KNIME Cluster Executor
Streaming in KNIME
Velocity

• High Demand Scoring/Prediction:
  – High Performance Scoring using generic Workflows
  – High Performance Scoring of Predictive Models

• Continuous Data Streams
  – Streaming in KNIME

• Distributed Computation
  – KNIME Cluster Executor
KNIME Cluster Executor: Distributed Data
KNIME Cluster Execution: Distributed Analytics
Volume
Moving computation to the data
Volume

• Database Extension
• Introduction to Hadoop
• KNIME Big Data Connector
• KNIME Spark Executor
Database Extension

- Visually assemble complex SQL statements
- Connect to almost all JDBC-compliant databases
- Harness the power of your database within KNIME
In-Database Processing

• Operations are performed within the database
Tip

- SQL statements are logged in KNIME log file
Database Port Types

Database JDBC Connection Port (light red)
• Connection information

Database Connection Port (dark red)
• Connection information
• SQL statement

Database Connection Ports can be connected to Database JDBC Connection Ports but not vice versa
Database Connectors

- Nodes to connect to specific Databases
  - Bundling necessary JDBC drivers
  - Easy to use
  - DB specific behavior/capability
- Hive and Impala connector part of the commercial KNIME Big Data Connectors extension
- General Database Connector
  - Can connect to any JDBC source
  - Register new JDBC driver via preferences page
Register JDBC Driver

- Open KNIME and go to File -> Preferences
- Increase connection timeout for long running database operations
- Register single jar file JDBC drivers
- Register new JDBC driver with companion files
Query Nodes

- Filter rows and columns
- Join tables/queries
- Extract samples
- Bin numeric columns
- Sort your data
- Write your own query
- Aggregate your data
Database GroupBy – Manual Aggregation

Returns number of rows per group
Database GroupBy – Pattern Based Aggregation

Tick this option if the search pattern is a regular expression otherwise it is treated as string with wildcards ("*" and '?")
Database GroupBy – Type Based Aggregation

- Matches all columns
- Matches all numeric columns
Database GroupBy – DB Specific Aggregation Methods

SQLite 7 aggregation functions

PostgreSQL 25 aggregation functions
Database GroupBy – Custom Aggregation Function
Database Writing Nodes

- Create table as select
- Insert/append data
- Update values in table
- Delete rows from table
Performance Tip

– Increase batch size in database manipulation nodes

Increase batch size for better performance
Volume

- Database Extension
- Introduction to Hadoop
- KNIME Big Data Connector
- KNIME Spark Executor
Apache Hadoop

• Open-source framework for distributed storage and processing of large data sets
• Designed to scale up to thousands of machines
• Does not rely on hardware to provide high availability
  – Handles failures at application layer instead
• First release in 2006
  – Rapid adoption, promoted to top level Apache project in 2008
• Spawned diverse ecosystem of products
Hadoop Ecosystem

- **Access**: HIVE
- **Processing**: MapReduce, Tez, Spark
- **Resource Management**: YARN
- **Storage**: HDFS
HDFS

- Hadoop distributed file system
- Stores large files across multiple machines

File

Blocks (default: 64MB)

DataNodes

File (large!)
HDFS – NameNode and DataNode

**NameNode**
- Master server that manages file system namespace
  - Maintains metadata for all files and directories in filesystem tree
  - Knows on which datanode blocks of a given file are located
- Whole system depends on availability of NameNode

**DataNodes**
- Workers, store and retrieve blocks per request of client or namenode
- Periodically report to namenode that they are running and which blocks they are storing
Reading Data from HDFS

HDFS Client

1: open
3: read
6: close

Distributed FileSystem

2: get block locations

FSDataInputStream

NameNode

4: read
5: read

DataNode

DataNode

DataNode
HDFS – Data replication and file size

Data Replication

• All blocks of a file are stored as sequence of blocks

• Blocks of a file are replicated for fault tolerance (usually 3 replicas)
  – Aims: improve data reliability, availability, and network bandwidth utilization
HDFS – Access and File Size

- Several ways to access HDFS data
  - FileSystem (FS) shell commands
    - Direct RPC connection
    - Requires Hadoop client to be installed
  - WebHDFS
    - Provides REST API functionality, lets external applications connect via HTTP
    - Direct transmission of data from node to client
    - Needs access to all nodes in cluster
  - HttpFS
    - All data is transmitted to client via one single node -> gateway

File Size

- Hadoop is designed to handle fewer large files instead of lots of small files
- Small file: File significantly smaller than Hadoop block size

Problems:
- Namenode memory
- MapReduce performance
YARN

• Cluster resource management system
• Two elements
  – Resource manager (one per cluster):
    • Knows where workers are located and how many resources they have
    • Scheduler: Decides how to allocate resources to applications
  – Node manager (many per cluster):
    • Launches application containers
    • Monitor resource usage and report to Resource Manager
YARN
Hive

- Infrastructure on top of Hadoop
- Provides data summarization, query, and analysis
- SQL-like language (HiveQL)
- Converts queries to MapReduce, Apache Tez, and Spark jobs
- Supports various file formats:
  - Text/CSV
  - SequenceFile
  - Avro
  - ORC
  - Parquet
Spark

• Cluster computing framework for large-scale data processing
• Keeps large working datasets in memory between jobs
  – No need to always load data from disk -> way (!) faster than MapReduce
• Great for:
  – Iterative algorithms
  – Interactive analysis
Spark – Basic Concepts

• SparkContext
  – Main entry point for Spark functionality
  – Represents connection to a Spark cluster
  – Create RDDs, accumulators, and broadcast variables on cluster

• RDD: Resilient Distributed Dataset
  – Read-only multiset of data items distributed over cluster of machines
  – Fault-tolerant: Lost partition automatically reconstructed from RDDs it was computed from
  – Lazy evaluation: Computation only happens when action is required
Spark – DataFrame and Dataset

• DataFrame
  – Distributed collection of data organized in named columns
  – Similar to table in relational database
  – Can be constructed from many sources: structured data files, Hive table, RDDs...

• Dataset
  – Extension of DataFrame API
  – Strongly-typed, immutable collection of objects mapped to a relational schema
  – Catches syntax and analysis errors at compile time
Volume

• Database Extension
• Introduction to Hadoop
• KNIME Big Data Connector
• KNIME Spark Executor
KNIME Big Data Connectors

• Package required drivers/libraries for HDFS, Hive, Impala access

• Preconfigured connectors
  – Hive
  – Cloudera Impala
  – Extends the open source database integration
Hive/Impala Loader

- Batch upload a KNIME data table to Hive/Impala
HDFS File Handling

• New nodes
  – HDFS Connection
  – HDFS File Permission

• Utilize the existing remote file handling nodes
  – Upload/download files
  – Create/list directories
  – Delete files
HDFS File Handling

Upload csv file to HDFS file system

Data Generator → CSV Writer → Variable to Table Row → String to URI → Upload

Download csv file from HDFS file system

List Remote Files → Create Temp Dir to Variable → Table Row to Variable → Delete Files

connection info

the file name

Delete Files

delete the remote file
Volume

- Database Extension
- Introduction to Hadoop
- KNIME Big Data Connector
- KNIME Spark Executor
KNIME Spark Executor

- Based on Spark MLlib
- Scalable machine learning library
- Runs on Hadoop
- Algorithms for
  - Classification (decision tree, naïve bayes, ...)
  - Regression (logistic regression, linear regression, ...)
  - Clustering (k-means)
  - Collaborative filtering (ALS)
  - Dimensionality reduction (SVD, PCA)
Familiar Usage Model

• Usage model and dialogs similar to existing nodes
• No coding required
MLlib Integration

- MLlib model ports for model transfer
- Native MLlib model learning and prediction
- Spark nodes start and manage Spark jobs
  - Including Spark job cancelation
Data Stays Within Your Cluster

- Spark RDDs as input/output format
- Data stays within your cluster
- No unnecessary data movements
- Several input/output nodes e.g. Hive, hdfs files, ...
Machine Learning – Unsupervised Learning Example
Machine Learning – Supervised Learning Example
Mass Learning – Fast Event Prediction

- Convert supported MLlib models to PMML
Sophisticated Learning - Mass Prediction

• Supports KNIME models and pre-processing steps
Closing the Loop

Apply model on demand

Sophisticated model learning

PMML model

Spark MLlib to PMML

Spark PMML Model Predictor

Learn model at scale

MLlib model

Apply model at scale
Mix and Match

• Combine with existing KNIME nodes such as loops
Modularize and Execute Your Own Spark Code
Lazy Evaluation in Spark

- Transformations are lazy
- Actions trigger evaluation
Spark Node Overview
KNIME Big Data Architecture

Scheduled execution and RESTful workflow submission

KNIME Server with extensions:
- KNIME Big Data Connectors
- KNIME Big Data Executor for Spark

Workflow Upload via HTTP(S)

Build Spark workflows graphically

KNIME Analytics Platform with extensions:
- KNIME Big Data Connectors
- KNIME Big Data Executor for Spark

Submit Impala queries via JDBC

Submit Hive queries via JDBC

Submit Spark jobs via HTTP(S)

*Software provided by KNIME, based on https://github.com/spark-jobserver/spark-jobserver
Executing KNIME Nodes on Spark
Behind the Scene

Execute KNIME workflow on Spark

KNIME Analytics Platform
KNIME Server

Workflow Replica
Behind the Scene

• **Variation (1):** Send RDD data through a single workflow replica
Behind the Scene

- **Variation (2):** Send pre-grouped RDD data through workflow replicas
Big Data, IoT, and the three V

• Variety:
  – KNIME inherently well-suited: open platform
  – broad data source/type support
  – extensive tool integration

• Velocity:
  – High Performance Scoring of predictive models
  – Streaming execution

• Volume:
  – Bring the computation to the data
  – Big Data Extensions cover ETL and model learning
  – Distributed Execution of KNIME workflows
Demo
Want to try it at home?

• Hadoop cluster
  – Use your own Hadoop cluster
  – Use a preconfigured virtual machine
    • http://hortonworks.com/products/hortonworks-sandbox/
    • http://www.cloudera.com/downloads/quickstart_vms.html

• Download and install compatible Spark Job Server
  – See installation steps at https://www.knime.org/knime-spark-executor#install

• For a free 30-day Trial go to https://www.knime.org/knime-big-data-extensions-free-30-day-trial
Resources

- SQL Syntax and Examples ([www.w3schools.com](http://www.w3schools.com))
- **The KNIME Website** ([www.knime.org](http://www.knime.org))
  - Database Documentation ([https://tech.knime.org/database-documentation](https://tech.knime.org/database-documentation))
  - Forum ([tech.knime.org/forum](http://tech.knime.org/forum))
  - LEARNING HUB under RESOURCES ([www.knime.org/learning-hub](http://www.knime.org/learning-hub))
  - Blog for news, tips and tricks ([www.knime.org/blog](http://www.knime.org/blog))
- **KNIME TV** channel on [YouTube](http://www.youtube.com)
- **KNIME** on [Twitter](http://twitter.com) [@KNIME]
The KNIME® trademark and logo and OPEN FOR INNOVATION® trademark are used by KNIME.com AG under license from KNIME GmbH, and are registered in the United States.