Speaker Notes: Scala for Apache Spark

Markus Dale, medale@asymmetrik.com

Jan 2019

Setup

- Open Spark API:

  https://spark.apache.org/docs/latest/api/scala/index.html
- Open Scala API: https://www.scala-lang.org/api/2.11.8/#package
- Open Java API String: https://docs.oracle.com/javase/8/docs/api/

- Bio:
    - mostly Java, big data with Hadoop
    - big data with Spark, Databricks, Scala
    - Now Asymmetrik - Scala, Spark, Elasticsearch, Akka...
    - Data Engineer
- Slides: https://github.com/medale/scala-spark/blob/master/presentation/ScalaSpark.pdf
- Scala Spark Code Examples: https://github.com/medale/scala-spark
- Also https://github.com/medale/spark-mail

- Intro to Scala (from Java) to leverage Apache Spark with Scala API
- sbt - build tool
- spark-testing framework for integration testing

- Data Engineer - scalable ecosystem of Java/Scala-based tools
- less boilerplate/less typing (Ted Malaska (three big data books on O'Reilly): 50% less than Java)
- strong typing, elegant multi-paradigm language (functional and OO)
- all code runs in executor JVM - no callouts to local Python shell for UDFs/UDAFs
- spark-shell is Scala-based
- Baltimore Scala meetup

- semicolons
- get/set JavaBeans convention
- explicit constructor
- static main method

Scala Main One

- Look Ma - no semicolons
- package structure - match directory structure
- Match file name/class name
- object vs. class
    - object - Java static methods, singleton
- no public (default)
- def - method/function declaration
- type declared after variable name
- return type, body (last entry gets returned)
- val - immutable, var - mutable
- class constructor (args none vs. val vs. var)
    - Java get/set: `import scala.reflect.BeanProperty`
    - `@BeanProperty var firstName`

- println statements
- default class to String - fully qualified class name@...

- immutable data structure
- default constructor parameters are `val`
- generates boiler plate code, singleton object
- Scala: Lots of "syntactic sugar" - less typing, compiler translates

- toString method created, prints class name, instance variables

- javap disassembler (package names removed)
- implements Product (abstract algebraic type), Serializable
    - Serializable important for Spark shuffle!
    - Product
        - productArity
        - productElement(int)
        - productIterator
- static apply factory method/unapply for matching
- copy method
- equals, hashCode, toString (see javap output next)

- expressions returns value (vs. statements)
    - Array (any indexed sequence) accessor args(index)
    - type of lines is inferred as lowest-common denominator if/else block

- Main division - AnyVal vs. AnyRef (unified through Any)
- AnyRef is like Object in Java
    - Null is a subclass of all reference classes
- AnyVal - Java primitives
    - Unit - `val u = ()` //u: Unit = () 0-tuple
    - Value class: `class Wrapper(val underlying: Int) extends AnyVal`
- Universal trait:
    - trait that extends Any
    - only has defs as members
- Nothing is subclass of everything (throwing exception returns Nothing)

- object - main method entry point
- SparkSession Scala API
    - scaladocs
    - object
    - class
    - .builder method (don't need empty parentheses - mutator method with ())
- Builder class - fluent interface/method chaining
    - getOrCreate
    - Run from shell or batch spark-submit

- Showed SparkSession scaladocs

- Triple quoted string can include special chars like newline, double-quote
- .stripMargin - by default uses pipe | removes all chars in front of pipe
- String - where does stripMargin method come from?

- java.lang.String - does not have stripMargin method

- `implicit def augmentString(x: String): StringOps`
- Also: `implicit def booleanArrayOps(xs: Array[Boolean]):`
  `ArrayOps[Boolean]`

- stripMargin method
- Also useful: head, tail, map, filter, sliding (ngrams), permutations

Want to cover - highlights but have in-depth examples in repo

- Intellij Scala plugin
- object/main method
- case class - Product
- functions - defining a function, anonymous functions
- collections - map, flatMap, filter
- immutability
- implicits - Predef / StringOps / StringLike
- Scala docs
- Spark - RDD, Dataframe, Dataset (Tungsten memory, code gen)
- SparkSession, DataframeReader
- udf
- sbt build - quick overview
- integration testing - spark-testing-base