CTIC UNI
CENTRO DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES
UNIVERSIDAD NACIONAL DE INGENIERÍA

CIENCIA DE DATOS:

APRENDE LOS FUNDAMENTOS DE MANERA PRÁCTICA

SESION 06

Scala

Juan Chipoco
mindquasar@gmail.com

*Las imágenes y graficos no me pertenecen y son usados solo con fines educativos

ÍNDICE

OBJETIVO **4**
SCALA **5**
FEATURES **8**
SCALA FOR APACHE SPARK **10**

OBJETIVO

Scala, abreviatura de Scalable Language, es un lenguaje de programación funcional híbrido. Fue creado por Martín Odersky. Scala integra sin problemas las características de los lenguajes funcionales y orientados a objetos. Scala está compilado para ejecutarse en la máquina virtual de Java. Muchas empresas existentes, que dependen de Java para aplicaciones comerciales críticas, están recurriendo a Scala para aumentar su productividad de desarrollo, escalabilidad de aplicaciones y confiabilidad general.

Esta sesion de introducción a Scala los ayudará a comprender los conceptos básicos del lenguaje de programación Scala. Aprenderá sobre Introducción a Scala, Historia de Scala, características de Scala, por qué Scala es popular y preferido sobre otros lenguajes de programación, diferencias entre Java y Scala.

Scala

¿Qué es la Programación Scala?

Scala es comparativamente nuevo en la escena de la programación, pero se ha vuelto popular muy rápidamente. Las siguientes declaraciones con grandes nombres muestran la popularidad de Scala en la industria:

Si tuviera que seleccionar un idioma para usar que no sea el lenguaje de programación Java, sería Scala. – James Gosling, creador de Java

Si hubiera visto el libro 'Programación en Scala' en 2003, probablemente nunca hubiera creado Groovy – James Strachan, creador de Groovy

Nada menos que la programación de Scala puede verse como un "reemplazo de Java", y el impulso detrás de Scala ahora es incuestionable: Charles Nutter, cocreador de Jruby

Scala Programming es un lenguaje de propósito general que combina conceptos de lenguajes de programación orientados a objetos y funcionales. Fue desarrollado para superar los problemas que enfrentan otros lenguajes y puede integrarse fácilmente en el código existente.

Historia de Scala

Scala Programming fue concebida por primera vez en 2001 en la École Polytechnique Fédérale de Lausanne por Martin Odersky, quien fue co-creador de Generic Java, javac y el lenguaje de programación Funnel de EPFL. El primer lanzamiento público de Scala se produjo en 2004, al que siguió la versión 2.0 en marzo de 2006. En 2012, recibió el premio al ganador del concurso ScriptBowl en la conferencia JavaOne.

Características de Scala

Algunas de las características clave de la programación de Scala incluyen:

- Es un lenguaje orientado a objetos que admite muchos patrones de diseño tradicionales heredados de los lenguajes de programación existentes.
- Es compatible con la programación funcional que le permite manejar la concurrencia y la programación distribuida en el nivel fundamental.

- Diseño de Scala para ejecutarse en una plataforma JVM que ayuda indirectamente a usar bibliotecas Java y otras API ricas en funciones.
- La programación de Scala está tipada estáticamente, lo que evita problemas de tipificación dinámica.
- Es fácil de implementar en proyectos Java existentes, ya que las bibliotecas Scala pueden usar dentro del código Java.
- No es necesario declarar variables en Scala, ya que el compilador de Scala puede inferir la mayoría de los tipos de variables.
- Se designan múltiples rasgos para una clase y luego se pueden combinar su interfaz y comportamiento.

¿Por qué Scala es popular?

Una de las razones clave del éxito de Scala es su estrecha integración con Java. Diseño de código fuente de Scala de manera que su compilador pueda interpretar las clases de Java y pueda utilizar completamente las bibliotecas, los marcos y las herramientas de Java.

Después de la compilación, los programas Scala pueden ejecutarse en máquinas virtuales Java (JVM) y Android. Para proyectos de desarrollo basados en web, Scala Programming puede incluso compilar a JavaScript.

Sin embargo, la programación de Scala es mucho más que una alternativa a Java. Es un lenguaje más conciso que utiliza una sintaxis simple y fácil de leer y requiere solo una fracción de las líneas de código en comparación con un programa típico de Java. Esto hace que la codificación de Scala sea más rápida y facilita las pruebas.

Debido a las razones anteriores, empresas como Linkedin, Twitter, Coursera, Foursquare, etc. han trasladado la mayoría de sus bases de código a Scala Programming. Muchos proyectos de código abierto como Apache Spark, Apache Kafka, etc. usan Scala para su Core.

Requisitos previos para aprender Programación en Scala
La programación de Scala se basa en Java, por lo que si tiene conocimiento de la sintaxis de Java, entonces es bastante fácil aprender Scala. Pero si no conoces Java pero conoces cualquier otro lenguaje de programación como C, C++ o Python, también puedes aprender Programación Scala muy rápidamente.

Scala vs. Java

Scala Programming tiene un conjunto de características que lo hacen completamente diferente del lenguaje Java.

- Todos los tipos son objetos.
- Tipo de inferencia

- Funciones anidadas
- Las funciones son objetos.
- Compatibilidad con lenguaje específico de dominio (DSL)
- Rasgos
- Cierres
- Soporte de concurrencia inspirado en Erlang

Features

Hay muchas características que lo hacen diferente de otros idiomas.

- Orientado a objetos: cada valor en Scala es un objeto, por lo que es un lenguaje de programación puramente orientado a objetos. El comportamiento y el tipo de objetos están representados por las clases y rasgos en Scala.
- Funcional: también es un lenguaje de programación funcional ya que cada función es un valor y cada valor es un objeto. Proporciona soporte para funciones de alto orden, funciones anidadas, funciones anónimas, etc.
- Estáticamente tipificado: el proceso de verificar y hacer cumplir las restricciones de los tipos se realiza en tiempo de compilación en Scala. A diferencia de otros lenguajes de programación tipificados estáticamente como C++, C, etc., Scala no espera la información de tipo redundante del usuario. En la mayoría de los casos, el usuario no necesita especificar un tipo.
- Extensible: se pueden agregar nuevas construcciones de lenguaje a Scala en forma de bibliotecas. Scala está diseñado para interpolar con JRE (Java Runtime Environment).
- Procesamiento concurrente y sincronizado: Scala permite al usuario escribir los códigos de una manera inmutable que facilita la aplicación del paralelismo (sincronización) y la concurrencia.
- Ejecutar en JVM y puede ejecutar código Java: Java y Scala tienen un entorno de tiempo de ejecución común. Así, el usuario puede pasar fácilmente de Java a Scala. El compilador de Scala compila el programa en un archivo .class, que contiene el código de bytes que JVM puede ejecutar. Todas las clases de Java SDK pueden ser utilizadas por Scala. Con la ayuda de Scala, el usuario puede personalizar las clases de Java.

Ventajas:

- Las características complejas de Scala proporcionaron una mejor codificación y eficiencia en el rendimiento.
- Tuplas, macros y funciones son los avances en Scala.
- Incorpora la programación funcional y orientada a objetos que a su vez lo convierten en un lenguaje poderoso.

- Es altamente escalable y, por lo tanto, brinda un mejor soporte para las operaciones de back-end.
- Reduce el riesgo asociado con la seguridad de subprocesos, que es mayor en Java.
- Debido al enfoque funcional, generalmente, un usuario termina con menos líneas de códigos y errores, lo que resulta en una mayor productividad y calidad.
- Debido al cálculo perezoso, Scala calcula las expresiones solo cuando son necesarias en el programa.
- No hay métodos estáticos ni variables en Scala. Utiliza el objeto singleton (clase con un objeto en el archivo fuente).
- También proporciona el concepto Traits. Los rasgos son la colección de métodos abstractos y no abstractos que se pueden compilar en las interfaces de Java.

Desventajas:

- A veces, dos enfoques hacen que la Scala sea difícil de entender.
- Hay un número limitado de desarrolladores de Scala disponibles en comparación con los desarrolladores de Java.
- No tiene optimización recursiva de cola verdadera ya que se ejecuta en JVM.
- Siempre gira en torno al concepto orientado a objetos porque cada función es un valor y cada valor es un objeto en Scala.

Aplicaciones:

- Se utiliza sobre todo en el análisis de datos con Spark.
- Se utiliza para desarrollar las aplicaciones web y la API.
- Proporciona la facilidad para desarrollar los marcos y las bibliotecas.
- Se prefiere usar en operaciones de back-end para mejorar la productividad de los desarrolladores.
- El procesamiento por lotes en paralelo se puede realizar con Scala.

Scala for Apache Spark

Lo más difícil para los desarrolladores de big data hoy en día es elegir un lenguaje de programación para aplicaciones de big data. La programación Python y R son los lenguajes elegidos entre los científicos de datos para construir modelos de aprendizaje automático, mientras que Java sigue siendo el lenguaje de programación de referencia para desarrollar aplicaciones Hadoop. Con la llegada de varios frameworks de big data como Apache Kafka y Apache Spark, el lenguaje de programación Scala ha ganado prominencia entre los desarrolladores de big data.

Con soporte para múltiples lenguajes de programación como Java, Python, R y Scala en Spark, a menudo se vuelve difícil para los desarrolladores decidir qué lenguaje elegir cuando trabajan en un proyecto de Spark. Una pregunta común que se les hace a los expertos de la industria en ProjectPro es: "¿Qué lenguaje debo elegir para mi próximo proyecto de Apache Spark?" La respuesta a esta pregunta varía, ya que depende de la experiencia en programación de los desarrolladores, pero preferiblemente el lenguaje de programación Scala se ha convertido en el lenguaje de elección para trabajar con marcos de big data como Apache Spark y Kafka.

¿Por qué debería aprender Scala para Apache Spark?

“Tener éxito en cualquier proyecto de Spark, ya sea su arquitectura, revisiones de código, mejores prácticas o soporte de producción, requiere la mejor experiencia del mundo”. – dijo Jamie Allen, director sénior de servicios globales de Typesafe.

El lenguaje de programación Scala, desarrollado por el fundador de Typesafe, brinda la confianza para diseñar, desarrollar, codificar e implementar las cosas de la manera correcta al hacer el mejor uso de las capacidades proporcionadas por Spark y otras tecnologías de big data.

Siempre hay una mejor herramienta de programación para cada tarea. Cuando se trata de procesar big data y aprendizaje automático, la programación de Scala ha dominado el mundo de big data y he aquí por qué:

1) Apache Spark está escrito en Scala y, debido a su escalabilidad en JVM, la programación de Scala es el lenguaje de programación más utilizado por los desarrolladores de big data para trabajar en proyectos Spark. Los desarrolladores afirman que el uso de Scala ayuda a profundizar en el código fuente de Spark para que puedan acceder e implementar fácilmente las características más nuevas de Spark. La interoperabilidad de Scala con Java es su mayor atractivo, ya que los desarrolladores de Java pueden emprender fácilmente el camino del aprendizaje al comprender rápidamente los conceptos orientados a objetos.

2) La programación de Scala conserva un equilibrio perfecto entre productividad y rendimiento.

   La mayoría de los desarrolladores de big data tienen experiencia en programación Python o R. La sintaxis para la programación de Scala es menos intimidante en comparación con Java o C++. Para un nuevo desarrollador de Spark sin experiencia previa, es suficiente que conozca las colecciones de sintaxis básicas y lambda para volverse productivo en el procesamiento de big data usando Apache Spark. Además, el rendimiento logrado con Scala es mejor que muchas otras herramientas tradicionales de análisis de datos como R o Python. Con el tiempo, a medida que se desarrollan las habilidades de un desarrollador, se hace más fácil la transición de un código de programación imperativo a uno funcional más elegante para mejorar el rendimiento.

3) Las organizaciones quieren disfrutar del poder expresivo del lenguaje de programación dinámico sin tener que perder la seguridad del tipo: la programación de Scala tiene este potencial y esto se puede juzgar por sus crecientes tasas de adopción en la empresa.

4) Scala está diseñado teniendo en cuenta el paralelismo y la concurrencia para aplicaciones de big data. Scala tiene un excelente soporte integrado de concurrencia y bibliotecas como Akka que facilitan a los desarrolladores la creación de una aplicación verdaderamente escalable.

5) Scala colabora bien con el modelo de big data de MapReduce debido a su paradigma funcional. Muchos marcos de datos de Scala siguen tipos de datos abstractos similares que son consistentes con las API de recopilación de Scala. Los desarrolladores solo necesitan aprender las colecciones estándar y sería fácil trabajar con otras bibliotecas.

6) El lenguaje de programación Scala proporciona el mejor camino para construir aplicaciones de big data escalables en términos de tamaño de datos y complejidad del programa. Con soporte para estructuras de datos inmutables, para comprensiones, valores con nombre inmutables, Scala proporciona un soporte notable para la programación funcional.

7) La programación de Scala es comparativamente menos compleja a diferencia de Java. Una sola línea de código complejo en Scala puede reemplazar de 20 a 25 líneas de

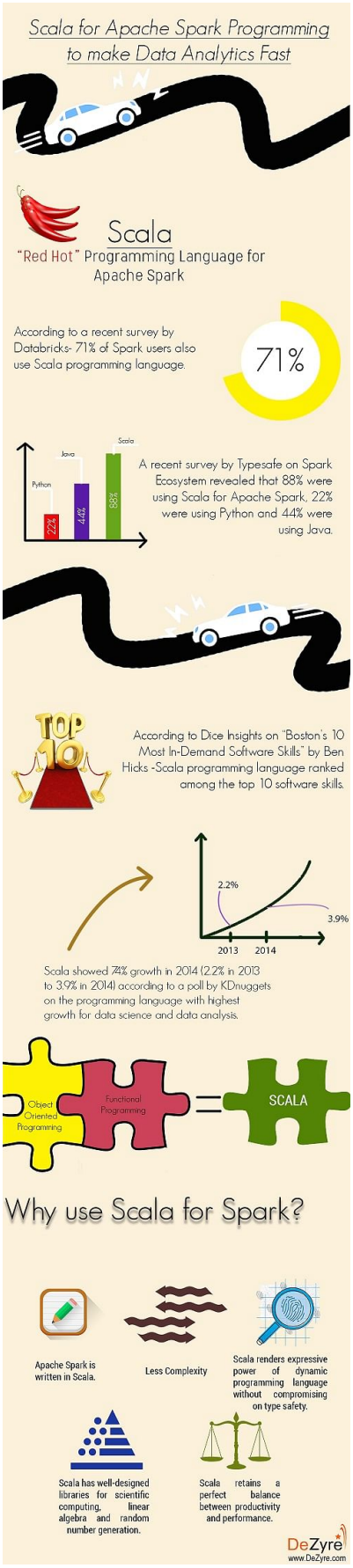
código Java complejo, lo que lo convierte en una opción preferible para el procesamiento de big data en Apache Spark.

8) Scala tiene bibliotecas bien diseñadas para computación científica, álgebra lineal y generación de números aleatorios. La biblioteca científica estándar Breeze contiene generación aleatoria no uniforme, álgebra numérica y otras funciones especiales. Saddle es la biblioteca de datos compatible con la programación de Scala que proporciona una base sólida para la manipulación de datos a través de estructuras de datos 2D, solidez a los valores faltantes, soporte respaldado por matrices y alineación automática de datos.

9) La eficiencia y la velocidad juegan un papel vital independientemente del aumento de la velocidad del procesador. Scala es rápido y eficiente, lo que lo convierte en una opción ideal de lenguaje para algoritmos computacionalmente intensivos. El ciclo de cómputo y la eficiencia de la memoria también están bien ajustados cuando se usa Scala para la programación de Spark.

10) Otros lenguajes de programación como Python o Java tienen retraso en la cobertura de la API. Scala ha cerrado esta brecha de cobertura de API y está ganando terreno en la comunidad de Spark. La regla general aquí es que al usar Scala o Python, los desarrolladores pueden escribir el código más conciso y al usar Java o Scala pueden lograr el mejor rendimiento en tiempo de ejecución. La mejor compensación es usar Scala para Spark, ya que hace uso de todas las funciones principales, en lugar de que los desarrolladores tengan que dominar las construcciones avanzadas.

La programación de Scala puede ser un lenguaje difícil de dominar para Apache Spark, pero vale la pena invertir el tiempo dedicado a aprender Scala para Apache Spark. Su combinación ganadora de paradigmas de programación orientada a objetos y funcional puede sorprender a los principiantes y es posible que les tome algún tiempo aprender la nueva sintaxis. La experiencia práctica en el trabajo con Scala para proyectos Spark es una ventaja adicional para los desarrolladores que desean disfrutar de la programación en Apache Spark sin complicaciones.


Scala for Apache Spark Programming
to make Data Analytics Fast
Scala
"Red Hot" Programming Language for
Apache Spark
According to a recent survey by
Databricks- 71% of Spark users also
use Scala programming language.
71%
Python
Java
Scala
22%
44%
88%
A recent survey by Typesafe on Spark
Ecosystem revealed that 88% were
using Scala for Apache Spark, 22%
were using Python and 44% were
using Java.
Top 10
According to Dice Insights on "Boston's 10
Most In-Demand Software Skills" by Ben
Hicks -Scala programming language ranked
among the top 10 software skills.
2.2%
3.9%
2013
2014
Scala showed 74% growth in 2014 (2.2% in 2013
to 3.9% in 2014) according to a poll by KDnuggets
on the programming language with highest
growth for data science and data analysis.
Object
Oriented
Programming
Functional
Programming
=
SCALA
Why use Scala for Spark?
Apache Spark is
written in Scala.
Less Complexity
Scala renders expressive
power of dynamic
programming language
without compromising
on type safety.
Scala has well-designed
libraries for scientific
computing, linear
algebra and random
number generation.
Scala retains a
perfect balance
between productivity
and performance.
DeZyre
www.DeZyre.com