UNIVERSIDAD SIMÓN BOLÍVAR

Scala en el desarrollo de la analítica en big data

Didier Roberto Acuña Urueta
CC 1192923615,
Código estudiantil: 201811292849
Correo: didier.acuna@unisimon.edu.co

Adrián Steven De La Rosa Althaona
CC 1002184011,
Código estudiantil: 201811294376
Correo: adrian.delarosa@unisimon.edu.co

Kevin Rafael García Salcedo
CC 1001780096,
Código estudiantil: 201811294335
Correo: kevingarcia030w@gmail.com

Adrián Gabriel Pico Martínez
CC 1143171233,
Código estudiantil: 201811293421
Correo: adrian.pico@unisimon.edu.co

Daniel Alfredo Rovira Hidrobo
CC 1140902254,
Código estudiantil: 201811294700
Correo: daniel.rovira@unisimon.edu.co

Trabajo de Investigación del Programa **de ingeniería de sistemas**

Tutores

Diana Judith Heredia Vizcaino

UNIVERSIDAD SIMÓN BOLÍVAR

RESUMEN:

Scala es uno de los lenguajes que últimamente se ha ido reconociendo poco a poco en las comunidades de desarrollo de habla inglesa, y ha ido agarrando fama en la analítica para el big data. Este lenguaje se ha diseñado para adaptarse a los requisitos futuros de los datos y, por lo tanto, se denomina un Lenguaje escalable (Scala). Scala no solo es un lenguaje totalmente orientado a objetos, sino también un lenguaje totalmente funcional para el análisis, clasificación y entre otros procesos funcionales de trata de datos.

Scala es un lenguaje que ya lleva más de una década desde que hizo su primera aparición, pero este se ha visto opacado por grandes lenguajes como Python, R, entre otros. Sin embargo, en los últimos años este ha comenzado a volverse más popular entre la comunidad de analistas y programadores, llevando su mayor enfoque y uso en el ámbito de la analítica big data, en el que tiene gran aplicabilidad debido a su estructuración y características.

Con esta investigación queremos determinar las ventajas, desventajas y funcionalidades que puede tener Scala, así mismo como comprobar qué tan eficiente es en el uso del big data, y cómo se desenvuelve en este mundo de análisis de volúmenes grandes datos. Para esto usaremos las metodologías del estudio y evaluación de criterios comparativos con el fin de evaluar la efectividad de Scala en el ámbito del big data. Por esto nuestros objetivos se enfocan en hallar la forma adecuada de solucionar, optimizar y mejorar el análisis en big data, soportado en el lenguaje de programación SCALA con sus variantes como lo es Spark-Apache, así como también en las diversas herramientas que ofrece este lenguaje para agilizar desarrollos y análisis de datos que en los lenguajes comunes no se podría. De esta forma se puede hacer la comparativa de SCALA con otros lenguajes que también son usados en el ámbito de la analítica big data, como lo es Python.

También queremos mostrar las diferencias entre los comandos mediante una comparativa entre Scala y su contra parte (Python), analizando un conjunto de datos similar y usando los mismos comandos, pero respectivos para cada lenguaje con el fin de ver cuál es más eficiente, fácil e intuitivo para hacer ciertas cosas en específico. Cada uno de estos lenguajes se harán en entornos de programación diferente, como: para Python usaremos Anaconda y extensión Jupyter-Notebook que nos ofrecerá ya un gran número de métodos preinstalados para este lenguaje, y para Scala se trabajará con la aplicación en la nube de Microsoft-Azure, con la herramienta Databricks para el manejo de Spark para Scala.

Se quiere a su parte hacer ver cómo un lenguaje como lo es Scala esta a la altura de leguaje de alto uso como lo es Python para el análisis, preprocesamiento y demás funciones de big data para un conjunto grande datos. Junto con esto traer en evidencia como Scala con su nuevo interprete que es Spark-Apache, y

conociendo que sigue siendo aún más viejo que Python, este pude competir con este grande de la big data, y tener la posibilidad de competir con este.

Antecedentes:

Construcción de hardware en un lenguaje integrado Scala
Al incorporar Chisel(El cual es un nuevo lenguaje de construcción de hardware que admite el diseño de hardware avanzado utilizando generadores altamente parametrizados y lenguajes de hardware específicos de dominio en capas.) en el lenguaje de programación Scala, se aumenta el nivel de abstracción del diseño de hardware al proporcionar conceptos que incluyen orientación a objetos, programación funcional, tipos parametrizados e inferencia de tipos. Chisel puede generar un simulador de software de alta velocidad con precisión de ciclo basado en C ++, o Verilog de bajo nivel diseñado para mapear ya sea en FPGA o en un flujo ASIC estándar para síntesis. Este artículo presenta Chisel, su incrustación en Scala, ejemplos de hardware y resultados para la simulación C ++, la emulación Verilog y la síntesis ASIC.

Rendimiento y eficiencia energética de Scala en dispositivos móviles
El objetivo de este estudio fue comparar cómo funcionan los nuevos lenguajes como Scala en entornos móviles en comparación con los lenguajes clásicos. Dado que Scala también se ejecuta en Java Virtual Machine (JVM), es posible ejecutar el código en dispositivos Android y compararlo con Java.

Simulación de tráfico urbano a gran escala con Scala y sistema informático de alto rendimiento/Large-scale urban traffic simulation with Scala and high-performance computing system
En este artículo de investigación se centra en los sistemas de información con máximo rendimiento que permiten simulación a gran escala, se expresa que los paradigmas anteriores desarrollaban soluciones eficaces y eficientes basados en plataformas populares de interfaz de paso de mensajes.
Se investiga la implementación escalable para un sistema de tráfico asincrónica, usando Scala/AKKA para la programación paralela y distribuida necesaria, además de su funcionalidad optima en sistema de información similares al estudiado.

BIG DATA: changes in data management
En este diario podemos apreciar la importancia de la aplicación de big data en el mundo y nos presenta la idea de como podemos aplicar los diferentes conocimientos como lenguajes de programación como SCALA para optimizar la interpretación y el manejo de esta gran cantidad de datos, para que con las conclusiones que nos de el programa, el analista de datos pueda crear un veredicto que afecte de forma positiva el entorno sobre el cual se estudio el dato y así mejorarlo.

UNIVERSIDAD SIMÓN BOLÍVAR

Técnicas big data: análisis de textos a gran escala para la investigación científica y periodística

En este artículo se nos presenta el análisis (basado en datamining, data science, big data y análisis sentimiento) de grandes cantidades de textos, para poder concluir lo que estos nos dicen y como estos datos pueden ser aplicados en el beneficio de la empresa, en donde existe la posibilidad de aplicar a este entorno el uso del lenguaje de programación SCALA en conjunto de la herramienta JUPYTER para lograr una versatilidad y mayor escalabilidad cuando de resultados exactos se trata.

Autómatas de datos en Scala

Este articulo trata de la temática de los autómatas incorporados en el mundo de los datos, así como también hablan del monitoreo y secuencia de eventos y como estos se realizan por medio de una API realizada con el lenguaje de programación SCALA.

Un estudio de analítica de Big Data usando Apache Spark con Python y Scala

En este estudio se hace una comparativa de la implementación en Apache Spark de los lenguajes Python y Scala y como estos se comportan ante los altos flujos y variabilidad de datos que se trabajan con Big Data. También se explica brevemente como es la estructura de clústeres en memoria de Apache Spark.

Objetivos:

Objetivo General

Evidenciar las ventajas y funcionalidades de Scala en el desarrollo de analítica de big data.

Objetivos específicos

- Identificar las funcionalidades del lenguaje Scala en el desarrollo de analítica de big data.
- Establecer criterios de evaluación en el desarrollo de analítica de big data.
- Realizar análisis comparativo entre Scala y Python en el desarrollo de analítica de big data por medio de un ejercicio práctico.

Materiales y Métodos:

Los siguientes párrafos exponen la metodologías requerida y usada para cumplir con cada uno de los objetivos propuestos, dando de esta manera el cumplimento del objetivo general de nuestra investigación

Fase 1: Identificar las funcionalidades del lenguaje Scala en el desarrollo de analítica de Big Data

Para realizar la fase 1 era menester realizar:

> Identificar y analizar diferentes estudios científicos de investigación, de desarrollo o programas académicos, en donde los expertos de Big Data utilicen Scala.
>
> Profundizar en el uso y aplicabilidad que posee Scala (analizar su lenguaje) para así generar una conclusión de formas en que Scala puede impactar un estudio de big data.
>
> Estableces una comparación entre las funcionalidades de escala para Big data en contra otros lenguajes (a priori, la profundización va adelante) usados para el mismo fin (Python).

Culminando de esta manera la fase 1 y dando inicio a la fase la cual

Fase 2: Establecer criterios de evaluación en el desarrollo de analítica de big data.

Para el desarrollo de esta fase se tuvieron en cuenta diferentes estudios científicos y académicos en los cuales consideran una comparativa entre lenguajes de programación, para así poder evidenciar cuales métricas generales de la programación podrían ser considerada en nuestra investigación, estableciendo al final de la fase las métricas evaluativas de desempeño y la escala de clasificación, donde la construcción de las métricas y escala de clasificación tienen en cuenta los requerimientos que se consideraron más importantes para la realización de la fase 3, la prueba piloto.

Fase 3: Realizar análisis comparativo entre Scala y Python en el desarrollo de analítica de big data por medio de un ejercicio práctico.

Para desarrollar la última fase se evaluaron los entornos de desarrollos para Python y Scala y se optaron por Anaconda para Python y Databricks para Scala, configurando los IDE’s para la implementación de prueba piloto en Scala y Python bajo las mismas condiciones con los requerimientos previamente establecidos.

Resultados:

Para facilidad del desarrollo de la investigación se comenzó el desarrollo de las pruebas pilotos bajo Python en Anaconda, dado que se consideró desde la experiencia de los investigadores, que era posible un rápido desarrollo en Python y de esta manera ceder más tiempo para el desarrollo en Scala bajo Databricks, en el cual la experiencia no era del mismo espectro y además es el eje central de la investigación.

Para brindar mas facilidad a la investigación, no se evaluaron todos los códigos o líneas de comandos, ya que era demasiado código para condesar la información, por esto se optó por 6 líneas de comandos que serian evaluadas bajos las métricas establecidas, en ambos lenguajes; Estos códigos evaluados son representante de la misma función en los diferentes lenguajes, esto quiere decir que, se evaluaron los códigos de leer datos en ambos lenguajes, por ejemplo.

Evaluadas las métricas evaluativas originales para cada lenguaje, se procedió a unir la evaluación para cada lenguaje en una sola tabla comparativa, de tal manera que facilitara la comprensión de la comparativa, y claramente evidenciar los resultados obtenido en base a la evaluación de las líneas de comando bajo las métricas evaluativas: Complejidad de comandos, Tiempo de ejecución y Funcionalidad resultante.

Conclusiones:

El objetivo de esta investigación está en evidenciar las ventajas y funcionalidades de Scala en el desarrollo de analítica de Big Data, el motivo de este análisis surge de la necesidad de la analítica de Big Data en la actualidad y con esta necesidad un gran campo de herramientas para trabajar bajo Big Data.

Gracias al desarrollo de las pruebas pilotos, en esta investigación podemos concluir que después de desarrollar las pruebas pilotos bajo herramientas y lenguajes diferentes, se puede apreciar que Scala tiene una funcionalidad de calidad tanto como Python, más sin embargo este gran potencial que tiene Scala se ve obstruido por la poca documentación y tutoriales de enseñanza que se tienen con este lenguaje enfocado en la Big Data.

Palabras clave: Lenguaje de programación Scala, Analitica Big data, Scala vs Python, prueba piloto scala-python.

UNIVERSIDAD SIMÓN BOLÍVAR

ABSTRACT:

Scala is one of the languages that lately has been gradually recognized in the English-speaking development communities, and has been gaining fame in analytics for big data. This language has been designed to accommodate future data requirements and is therefore called a Scalable Language (Scala). Scala is not only a fully object-oriented language, but also a fully functional language for analysis, classification and among other functional data processing processes.

Scala is a language that has been over a decade since it first appeared, but it has been overshadowed by great languages such as Python, R, among others. However, in recent years it has begun to become more popular among the community of analysts and programmers, taking its greater focus and use in the field of big data analytics, in which it has great applicability due to its structure and characteristics.

With this research we want to determine the advantages, disadvantages and functionalities that Scala can have, as well as to check how efficient it is in the use of big data, and how it operates in this world of analysis of large volumes of data. For this we will use the methodologies of the study and evaluation of comparative criteria in order to evaluate the effectiveness of Scala in the field of big data. For this reason, our objectives are focused on finding the appropriate way to solve, optimize and improve the analysis in big data, supported in the SCALA programming language with its variants such as Spark-Apache, as well as in the various tools that this offers. language to streamline development and data analysis that common languages could not. In this way, SCALA can be compared with other languages that are also used in the field of big data analytics, such as Python.

We also want to show the differences between the commands by comparing Scala and its counterpart (Python), analyzing a similar data set and using the same commands, but respective for each language in order to see which one is more efficient, easier and more efficient. intuitive to do specific things. Each of these languages will be made in different programming environments, such as: for Python we will use Anaconda and the Jupyter-Notebook extension that will already offer us a large number of pre-installed methods for this language, and for Scala we will work with the application in the cloud of Microsoft-Azure, with the Databricks tool for handling Spark for Scala.

You want to show how a language such as Scala is at the height of high-use language such as Python for analysis, preprocessing and other big data functions for a large data set. Along with this, bringing Scala into evidence with its new interpreter that is Spark-Apache, and knowing that it is still older than Python, it could compete with this big data great, and have the possibility of competing with it.

UNIVERSIDAD SIMÓN BOLÍVAR

Background:

Hardware construction in an embedded Scala language
By incorporating Chisel (which is a new hardware construction language that supports advanced hardware design using highly parameterized generators and layered domain-specific hardware languages.) Into the Scala programming language, the level of abstraction of the hardware design by providing concepts including object orientation, functional programming, parameterized types, and type inference. Chisel can generate a high-speed, loop-accurate C ++-based software simulator, or a low-level Verilog designed to map to either FPGA or standard ASIC stream for synthesis. This article presents Chisel, its Scala embedding, hardware examples, and results for C ++ simulation, Verilog emulation, and ASIC synthesis.

Scala performance and energy efficiency on mobile devices
The aim of this study was to compare how new languages like Scala work in mobile environments compared to classic languages. Since Scala also runs on the Java Virtual Machine (JVM), it is possible to run the code on Android devices and compare it to Java.

Large-scale urban traffic simulation with Scala and high-performance computing system
This research article focuses on information systems with maximum performance that allow large-scale simulation, it is stated that the previous paradigms developed effective and efficient solutions based on popular message passing interface platforms.
The scalable implementation for an asynchronous traffic system is investigated, using Scala / AKKA for the necessary parallel and distributed programming, in addition to its optimal functionality in information systems similar to the one studied.

BIG DATA:
changes in data management
In this journal we can appreciate the importance of the application of big data in the world and it presents us with the idea of how we can apply different knowledge such as programming languages such as SCALA to optimize the interpretation and handling of this large amount of data, so that With the conclusions that we receive from the program, the data analyst can create a verdict that positively affects the environment on which the data was studied and thus improve it.

Big data techniques:
large-scale text analysis for scientific and journalistic research
This article presents the analysis (based on datamining, data science, big data and sentiment analysis) of large amounts of texts, in order to conclude what they tell us and how these data can be applied to the benefit of the company, where there is the possibility of applying to this environment the use of the SCALA programming

language in conjunction with the JUPYTER tool to achieve versatility and greater scalability when it comes to exact results.

Data automata in Scala
This article deals with the subject of automata incorporated in the world of data, as well as talking about the monitoring and sequence of events and how these are carried out through an API made with the SCALA programming language.

A Big Data Analytics Study Using Apache Spark with Python and Scala
This study makes a comparison of the implementation in Apache Spark of the Python and Scala languages and how they behave in the face of the high flows and variability of data that are worked with Big Data. Apache Spark's in-memory cluster structure is also briefly explained.

Objectives:

General objective

Show the advantages and functionalities of Scala in the development of big data analytics.

Specific objectives

• Identify the functionalities of the Scala language in the development of big data analytics.

• Establish evaluation criteria in the development of big data analytics.

• Perform comparative analysis between Scala and Python in the development of big data analytics through a practical exercise.

Materials and methods:

The following paragraphs expose the methodologies required and used to meet each of the proposed objectives, thus fulfilling the general objective of our research.

Phase 1: Identify the functionalities of the Scala language in the development of Big Data analytics

To carry out phase 1 it was necessary to carry out:

Identify and analyze different scientific studies of research, development or academic programs, where Big Data experts use Scala.

Delve into the use and applicability of Scala (analyze its language) in order to generate a conclusion of ways in which Scala can impact a big data study.

You establish a comparison between the scale functionalities for Big data against other languages (a priori, the deepening goes ahead) used for the same purpose (Python).

Thus culminating phase 1 and starting the phase which

Phase 2: Establish evaluation criteria in the development of big data analytics.

For the development of this phase, different scientific and academic studies were taken into account in which they consider a comparison between programming languages, in order to show which general programming metrics could be considered in our research, establishing at the end of the phase the performance evaluation metrics and the classification scale, where the construction of the metrics and classification scale take into account the requirements that were considered most important for the realization of phase 3, the pilot test.

Phase 3: Perform comparative analysis between Scala and Python in the development of big data analytics through a practical exercise.

To develop the last phase, the development environments for Python and scala were evaluated and Anaconda for Python and Databricks for Scala were chosen, configuring the IDE's for the pilot test implementation in Scala and Python under the same conditions with the previously established requirements.

Results:

In order to facilitate the development of the research, the development of the pilot tests under Python in Anaconda began, since it was considered from the experience of the researchers, that a rapid development in Python was possible and thus allow more time for development in Scala under Databricks, in which the experience was not of the same spectrum and is also the central axis of the investigation.

To facilitate the investigation, not all the codes or command lines were evaluated, since it was too much code to condense the information, for this reason 6 command lines were chosen that would be evaluated under the established metrics, in both languages; These evaluated codes are representative of the same function in the different languages, this means that the codes for reading data in both languages were evaluated, for example.

Once the original evaluative metrics were evaluated for each language, the evaluation for each language was combined in a single comparative table, in such a way that it facilitated the understanding of the comparison, and clearly evidenced the results obtained based on the evaluation of the lines of command under the evaluative metrics: Command Complexity, Execution Time, and Resulting Functionality.

Conclusions:

The objective of this research is to show the advantages and functionalities of Scala in the development of Big Data analytics, the reason for this analysis arises from the need for Big Data analytics today and with this need a large field of tools to work under Big Data.

Thanks to the development of the pilot tests, in this research we can conclude that after developing the pilot tests under different tools and languages, it can be seen that Scala has quality functionality as well as Python, but nevertheless this great potential that Scala has is It is obstructed by the little documentation and teaching tutorials that are had with this language focused on Big Data.

KeyWords: Scala programming language, Big data analytics, Scala vs Python, scala-python pilot test.

UNIVERSIDAD SIMÓN BOLÍVAR

Referencias

[1] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis, J. Wawrzynek y K. Asanović, «construcción de hardware en un lenguaje integrado Scala,» 3-7 5 2012. [En línea]. Available: https://ezproxy.unisimon.edu.co:2131/document/6241660.

[2] M. Denti y J. Nurminen, «Rendimiento y eficiencia energética de Scala en dispositivos móviles,» *ScienceDirect,* Vols. %1 de %225-27, nº (25-27 de septiembre de 2013). Prhttps://ezproxy.unisimon.edu.co:2131/document/6658099/authors#authors., p. 9, 2013.

[3] M. Janczykowski, W. Turek, M. Malawski y A. Byrski, «Large-scale urban traffic simulation with Scala and high-performance computing system,» de *Journal of Computational Science*, 2019, pp. Pages 91-101.

[4] D. Sebalj, A. Živković y K. Hodak, «Big Data: Changes in Data Management,» *Ekomomski Vjesnik,* vol. 29, pp. 487-499, 2016.

[5] C. Arcila, E. Barbosa y F. Cabezuelo, «Técnicas big data: análisis de textos a gran escala para la investigación científica y periodística,» *El profesional de la información,* vol. 25, nº 4, pp. 626-631, 2016.

[6] K. Havelund, «Data Automata in Scala,» de *Conferencia sobre aspectos teóricos de la ingeniería de software*, 2014.

[7] Y. Gupta y S. Kumari, «A Study of Big Data Analytics using Apache Spark with Python and Scala,» *3rd International Conference on Intelligent Sustainable Systems (ICISS),* pp. 471-478, 3-5 Diciembre 2020.

[8] M. Odersky, «Artima,» 9 6 2006. [En línea]. Available: https://www.artima.com/weblogs/viewpost.jsp?thread=163733. [Último acceso: 30 3 2021].

[9] K. CODING, «Motivos para aprender scala,» 2018.

[10] K. CODING, «KEEP CODING,» 24 MAYO 2018. [En línea]. Available: https://keepcoding.io/blog/10-motivos-por-los-que-debes-aprender-scala/.

[11] M. Odersky, Programing in scala, mountain view california: PrePrintTM, 2007.

[12] M. Odersky, The programming scala specification, suiza: EPFL, 2011.

[13] «DOCS SCALA,» [En línea]. Available: https://docs.scala-lang.org/es/tour/tour-of-scala.html#:~:text=Scala%20es%20un%20lenguaje%20de,orientados%20a%20objetos%20y%20funcionales..

[14] «DCCIA,» [En línea]. Available: http://www.dccia.ua.es/dccia/inf/asignaturas/LPP/seminarios/seminario2-scala/seminario2-scala.html.

[15] «SCALA LANG,» [En línea]. Available: https://www.scala-lang.org/download/.

[16] Gary Briceño, «CLUB DE TECNOLOGIA,» 23 Abril 2015. [En línea]. Available: https://www.clubdetecnologia.net/blog/2015/scala-caracteristicas-del-codigo/.