Log clustering tool

# Resumen

Contenido

[Resumen i](#_Toc521001274)

[1. Introducción 1](#_Toc521001275)

[2. Herramientas y metodología. 2](#_Toc521001276)

[2.1 Python. 2](#_Toc521001277)

[2. Bibliografía 4](#_Toc521001278)

# 1. Introducción

Hoy en día se genera una gran cantidad de archivos de logs en campos como la industria y la ciencia, ya que estos son una fuente de información muy importante y útil en muchas situaciones, Sin embargo, a medida que aumenta la complejidad de los sistemas, el análisis de los archivos de logs es cada vez más exigente y dificultoso, ya que hay que realizar un gran esfuerzo para recopilar, almacenar e indexar una gran cantidad de logs que se agrava más cuando estos logs no son heterogéneos **(1)**.

Toda esta producción de logs hace, que, debido al tamaño de estos conjuntos de datos, que las soluciones de bases de datos convencionales no sean las adecuadas para el análisis de la información y en su lugar se consideren más apropiados bases de datos virtuales combinadas con sistemas de procesamiento distribuidos y paralelo.

Con todo esto, se puede suponer que el análisis de logs es un caso de uso de big data y por lo tanto es un gran desafío para su procesamiento, almacenamiento, variedad y su gestión con los recursos disponibles. Además, hay que añadir que cuando los logs provienen de múltiples fuentes surgen problemas con la extracción de contenido significativo y su correlación. Por todo esto surgen varias soluciones eficientes para tratar, reconocer y almacenar la información importante y que se puedan recuperar o migrar fácilmente entre los diferentes centros de datos.

En definitiva, el tratamiento de logs hoy en día tiene una gran importancia y por esta razón se han desarrollado muchos algoritmos para ello. Esto implica que tiene que existir computadoras muy potentes para dicho tratamiento, ya que una maquina genera muchos logs.

Con la monitorización de los archivos de logs se pueden detectar errores y/o anomalías en el funcionamiento de la máquina que genera los logs. Cuando se manifiesta un error, tiene que haber un experto que haya almacenado dicho error en una base de datos o algo similar, sin embargo, si el error que se ha generado no esta catalogado no puede ser detectado, pero llevar todo este trabajo a cabo lleva mucho tiempo y esfuerzo y además es propenso a que se cometan errores.

El objetivo de este trabajo es analizar logs a través de varios algoritmos de clusterización (que agrupa los logs en grupos y según sus patrón), para llevar a cabo estas tareas se trabajara con una base de datos no relacional (elasticsearch), con una herramienta de extracción, transformación y carga (logstash), con un framework de computación en clúster (apache spark) y con el lenguaje de programación Python y sus librerías, todas estas herramientas son open-source y están indicadas para trabajar con gran cantidad de datos.

# 2. Herramientas y metodología.

Para la realización de este trabajo se van a utilizar las herramientas que se describen a continuación.

## 2.1 Python.

Python es un lenguaje de programación interpretado que surgió en 1991 con la idea de que su sintaxis haga que el código sea legible más fácilmente y que en la actualidad tiene dos versiones estables que son la 3.7 y la 2.7 (que es la que se utiliza en este trabajo), la razón por la que aún se está dando soporte a la versión 2.7 es porque de esta versión se pasa a la 3.0 y hay un gran cambio cuando uno se pone a desarrollar su código o quiere realizar una migración.

Es un lenguaje de alto nivel, de uso general y que puede extenderse e incorporarse. Este lenguaje de programaci´on soporta varias filosof´ıas de programaci´on. Al tratarse de un lenguaje de programaci´on abierto, hay muchos paquetes en la nube que le dan una gran variedad de herramientas que hacen de Python un lenguaje muy extendido. Otra propiedad muy importante a la hora de programar en este lenguaje es que es lenguaje interpretado, lo que permite crear y ejecutar programas muy r´apido. Python, adem´as, es un lenguaje escrito en C, lo que permite sacar el m´aximo rendimiento a la m´aquina donde se ejecuta. Por otro lado, este lenguaje es capaz de gestionar la memoria utilizada sin que el programador se est´e preocupando constantemente por este problema. A la hora de programar, si programamos con una herramienta como Eclipse, 2 tenemos a nuestra dispocisi´on unas buenas herramientas de depuraci´on, lo que permite entender bien c´omo funciona el c´odigo y ser capaz no solo de localizar los errores, sino de comprender por qu´e se comete dicho error. Para m´as informaci´on, v´ease [5].

Ccc

Además, el análisis de los archivos de registro presenta algunos desafíos adicionales. Dado que muchos sistemas están distribuidos y son heterogéneos, los registros de una serie de componentes deben correlacionarse primero (Oliner et al., 2012). Además, tal vez haya datos faltantes, duplicados o engañosos que hacen que el análisis de registros sea más complejo o incluso imposible. Además, muchas técnicas de modelado analítico y estadístico no siempre proporcionan ideas accionables (Oliner et al., 2012).

ss (1)

# 3. Bibliografía

1. *Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark.* **Ilias Mavridis y Helen Karatza.** 2017, The Journal of Systems and Software, págs. 133-151.

2. **[En línea] https://spark.apache.org/.**