Análisis de datos mediante el algoritmo de clasificación J48, sobre un cluster en la nube de AWS

Data Analysis through the J48 classification algorithm, on AWS Cloud Cluster

Carlos Hernán Cardona Taborda[1]
Nancy Gelvez García[2]
Jairo Jamith Palacios Rozo[3]

Resumen

El siguiente artículo presenta la implementación del algoritmo J48 con el software libre Weka 3.8.0 ejecutado desde un clúster en la nube de AWS, el cual fue desarrollado con Starcluster 0.91. Este sistema es utilizado sobre una base de datos que contiene información de clasificación de vidrios a través del algoritmo, junto con datos de entrenamiento y validación cruzada, se logra crear un árbol de clasificación que permitirá predecir a qué clase de material pertenece el vidrio ingresado.

Palabras clave: análisis de datos, AWS, clúster, programación en la nube, J48, validación cruzada.

Abstract

The following article shows the J48 algorithm implementation using the open source software Weka 3.8.0 running from an AWS cloud cluster, which was developed on Starcluster 0.91. This system is used on a database that contains glass classification information through the algorithm. With training data and cross validation it is created a classification tree that allows predicting to which material the inserted glass belongs.

Keywords: AWS, cloud computing, cluster, cross validation, data analysis, J48.

1 carlos.cardona@exsis.com.co
2 nygelvezg@udistrital.edu.co
3 jjpalacios@unicolmayor.edu.co

Introducción

Uno de las tecnologías de más acogida los últimos años es la programación en la nube, la cual permite desarrollar tareas que eran imposibles hace 10 años, las empresas que utilizan programación en la nube le pagan a un proveedor de estos servicios como Amazon Web Service (AWS), para evitar los gastos de infraestructura, planeación y seguridad, esta les permite ejecutar sus servicios con una plataforma escalable que les da toda la capacidad que necesiten, es decir, se le cobra a las organizaciones por la capacidad de procesamiento que estas consuman.

Es una gran ventaja utilizar computación en la nube ya que la capacidad de procesamiento de estos servidores es casi ilimitada, estos proveedores soportan tecnologías como Netflix, que provee streaming de video a millones de personas alrededor del mundo con muy buena calidad todos los días de la semana a cualquier hora.

Las pequeñas empresas también empiezan a utilizar computación en la nube debido a la escalabilidad de los servicios, lo que se traduce como gastos a medida de sus necesidades y muchos menos gastos que si decidieran montar una infraestructura propia, hasta en el mismo costo de los computadores que solo deben tener una conexión banda ancha a internet que les permita utilizar los servicios, no deben tener gran capacidad de procesamiento como 10 años atrás.

Big data es un tema muy ligado a la programación en la nube, esta se refiere al almacenamiento y utilización las técnicas (algoritmos) que permiten analizar grandes cantidades de datos para solucionar problemas, se puede analizar desde lo que escriben las personas en las redes sociales hasta toma de muestras de aire para establecer la contaminación en un área y de que está compuesta.

Big data utiliza una serie de recursos para desarrollar sus análisis entre estos se encuentra:

Generados por usuarios: Son los datos generados a través de las redes sociales y otros portales públicos del internet.

Transacciones de datos: Estos se generan gracias a facturación, llamadas, o transacciones entre cuentas.

E-marketing: Generados a través de la navegación en la red que permite generar mapas de calor en las publicaciones o páginas que son más concurridas.

Machine to machine (M2M): Es la comunicación entre dispositivos como sensores de temperatura, luz, altura, presión, sonido y otros.

Biométrica: Estos datos provienen de las agencias de seguridad, salud y otras entidades gubernamentales del mundo.

En este artículo se reporta un caso que combina Big Data con programación en la nube utilizando los servicios de AWS, en un clúster que ejecuta Weka en una base de datos para establecer parámetros de rendimiento de estudiantes.

Utilizando una base de datos existente con registros de varios tipos de vidrio se desea establecer el tipo exacto de material teniendo en cuenta 7 tipos de vidrios de distintas fuentes de un vidrio. Se ingresa en el sistema, para esto se tienen en cuenta distintos factores que permiten discriminar el objeto por la composición química y la fuente de dicho objeto, así como se ve en la Tabla 1, propiedad de la UCI [15]. Dichos factores son:

[15] Tabla 1. Factores de medición

#		Nombre	Descripción
1		ID	Número de 1 a 124
2		RI	Índice de refracción
3		Na	Porcentaje de Sodio por mm
4		Mg	Porcentaje de magnesio por mm
5		Al	Porcentaje de Aluminio por mm
6		Si	Porcentaje de magnesio por mm
7		K	Porcentaje de magnesio por mm
8		Ca	Porcentaje de magnesio por mm
9		Ba	Porcentaje de magnesio por mm
10		Fe	Porcentaje de magnesio por mm
11		Tipo de vidrio (atributo de clase)	
11.1		Ventanas de construcción procesadas	
11.2		Ventanas de construcción sin procesar	
11.3		Ventanas de vehículo procesadas	
11.4		Ventanas de vehículos sin procesar	
11.5		Conteiner	
11.6		Vajilla	
11.7		Bombillos	

Para contextualizar este artículo, es necesario definir los siguientes conceptos:

Clúster

Un cúmulo, granja o clúster de computadoras, se puede definir como un sistema de procesamiento paralelo o distribuido. Consta de un conjunto de computadoras independientes, interconectadas entre sí, de tal manera que funcionan como un solo recurso computacional, según Hernández, Santillán y Caballero [16]. A cada uno de los elementos del

clúster se le conoce como nodo. Estos son aparatos o torres que pueden tener uno o varios procesadores, memoria RAM, interfaces de red, dispositivos de entrada y salida, y sistema operativo. Los nodos pueden estar contenidos e interconectados en un solo gabinete, o, como en muchos casos, acoplados a través de una red de área local LAN (Local Area Network). Otro componente básico en un clúster es la interfaz de la red, la cual es responsable de transmitir y recibir los paquetes de datos, que viajan a través de la red entre los nodos. Finalmente el lograr que todos estos elementos funcionen como un solo sistema, es la meta a la que se quiere llegar para dar origen a un clúster, según Hernández [4].

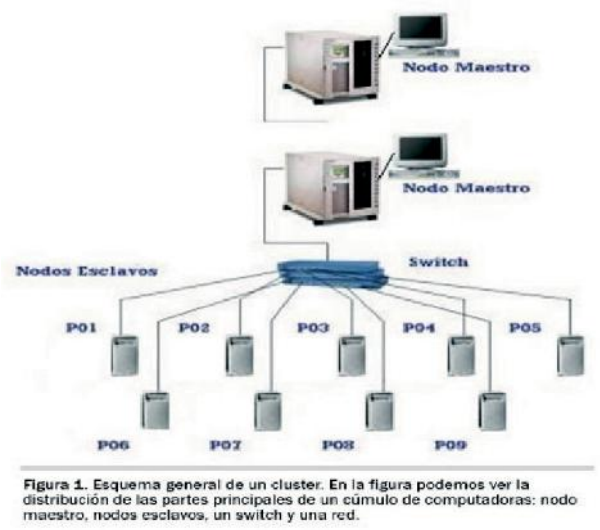


Figura 1. Esquema general de un cluster. En la figura podemos ver la distribución de las partes principales de un cúmulo de computadoras: nodo maestro, nodos esclavos, un switch y una red.

[16] Imagen 1. Esquema general de un clúster

Minería de datos

En los últimos años se han acumulado enormes cantidades de datos en todas las organizaciones, y esta tendencia continúa a un ritmo acelerado.
Esto es posible por el amplio uso de los sistemas computarizados, nuevas técnicas de captura de datos, el empleo de códigos de barra, los lectores de caracteres ópticos, las tarjetas magnéticas, entre otros, y por el avance en la tecnología de almacenamiento y su consiguiente reducción de costos. La disponibilidad de esos datos es un importante activo para cualquier organización, en la medida en que puedan ser transformados en información de interés, utilizando técnicas y métodos de Data Mining.

Data Mining, también referenciado como Descubrimiento del Conocimiento en Bases de Datos (Knowledge Discovery in Databases o KDD), ha sido definida como el proceso de extracción no trivial de información implícita, previamente desconocida y potencialmente útil.

El crecimiento explosivo de las bases de datos, de Internet y el empleo de técnicas y herramientas (que en forma automática y eficiente, generan información a partir de los datos almacenados), permiten descubrir patrones, relaciones y formular modelos. En particular, estas técnicas han adquirido enorme importancia en áreas tales como estrategias de marketing, soporte de decisiones, planeamiento financiero, análisis de datos científicos, bioinformática, análisis de textos y de datos de la web.

Data Mining incluye áreas del conocimiento tales como Estadística, Inteligencia Artificial (Machine Learning) y Bases de Datos. Se estima que del análisis de esos datos pueden surgir ventajas competitivas o novedosas soluciones a antiguos problemas. Data Mining y Knowledge Discovery es un área de gran actividad a nivel académico, como lo demuestran el gran número de eventos científicos relacionados, como así también laborales, acorde a la UBA [5].

Algoritmo J48

El algoritmo J48 implementado en Weka es una versión del clásico algoritmo de árboles de decisión C4.5 propuesto por Quilan. Los árboles de decisión entran dentro de los métodos de clasificación supervisada, es decir, se tiene una variable dependiente o clase, y el objetivo del clasificador es determinar el valor de dicha clase para casos nuevos. El proceso de construcción del árbol comienza por el nodo raíz, el que tiene asociados todos los ejemplos o casos de entrenamiento. Lo primero es seleccionar la variable o atributo a partir de la cual se va a dividir la muestra de entrenamiento original (nodo raíz), buscando que en los subconjuntos generados haya una mínima variabilidad respecto a la clase. Este proceso es recursivo, es decir, una vez que se haya determinado la variable con la que se obtiene la mayor homogeneidad respecto a la clase en los nodos hijos, se vuelve a realizar el análisis para cada uno de los nodos hijos. Aunque en el límite este proceso se detendría cuando todos los nodos hojas contuvieran casos de una misma clase, no siempre se desea llegar a este extremo, para lo cual se implementan métodos de pre-poda y post-poda de los árboles.

El algoritmo J48 amplía las funcionalidades del C4.5, tales como permitir la realización del proceso

de post-poda del árbol mediante un método basado en la reducción del error (reducedErrorPruning) o que las divisiones sobre las variables discretas sean siempre binarias (binarySplits), de acuerdo a Rivera, Rosete y Rodríguez [6].

Validación cruzada

La validación cruzada es una herramienta estándar de análisis que resulta muy útil a la hora de desarrollar y ajustar los modelos de minería de datos. La validación cruzada se usa después de crear una estructura de minería de datos y los modelos de minería de datos relacionados para determinar la validez del modelo. La validación cruzada tiene las siguientes aplicaciones:

- Validar la solidez de un modelo de minería de datos determinado.
- Evaluar varios modelos de una instrucción única.
- Generar varios modelos e identificar a continuación el mejor modelo basándose en estadísticas.
- En esta sección se describe cómo usar las características de validación cruzada proporcionadas para la minería de datos y cómo interpretar sus resultados para un único modelo o para varios basados en un único conjunto de datos, de acuerdo a Microsoft [7].

Se dividen las instancias en tantas carpetas como indica el parámetro folds, y encada evaluación se toman las instancias de cada carpeta como datos de prueba, el resto como datos de entrenamiento para construir el modelo. Los errores calculados serán el promedio de todas las ejecuciones, según García y Álvarez [8].

Dentro de los antecedentes, encontramos que con la promesa en la demanda de recursos de cálculo/almacenamiento, muchos usuarios están desplegando aplicaciones científicas intensivas de datos en la nube. Para acelerar estas aplicaciones, la posibilidad de almacenar en caché los datos intermedios mediante el cálculo elástico y el marco de almacenamiento ha demostrado ser prometedor.

Con este fin, se cree que un estudio en profundidad de las decisiones de ubicación de caché a través de varias opciones de almacenamiento de la nube sería altamente beneficioso para una gran clase de usuarios. Aunque se han propuesto análisis tangenciales, la nuestra por el contrario se centra en soluciones de compromiso coste-rendimiento de mantener una caché de datos con varios parámetros de cualquier aplicación en la nube. Se han comparado varios recursos de servicio Web de Amazon (AWS) como posibles ubicaciones de memoria caché y se encontró que los atributos dependientes de aplicaciones como el tamaño de unidad de datos, el tamaño total de la memoria caché, y la persistencia, influyen poderosamente en el costo de sustento caché.

Por otra parte, mientras que las memorias caché basados en instancia de esperar rendimiento más alto costo, el rendimiento que ofrecen pueden ser mayores opciones de menor costo, como dicen Chiu y Agrawal [10].

Ahora, observemos que se ha aportado desde el análisis de datos por medio de diferentes algoritmos ya establecidos:
Cloud computing data mining to SCADA for energy management

En este trabajo, se presenta la infraestructura de minería en la nube basado en SAP HANA para SCADA. Esto proporciona una decisión tomada poderosa herramienta para el operador en el centro de control de energía. Algunas de las características importantes de hardware y software de HANA están definidos. Se presenta una configuración del sistema SCADA para gestión de la energía, basada en la minería computación en la nube. También se describe la nube características y servicios de minería de datos del modelo de nube, de acuerdo a Gupta, Moinuddin y Kumar [11].

FSBD: A Framework for Scheduling of Big Data Mining in Cloud Computing

La computación en la nube es vista como una tecnología emergente para la minería de datos y análisis. La computación en nube puede proporcionar resultados de minería de datos en forma de un software como servicio (SAS). El rendimiento y la calidad de la minería son criterios fundamentos para el uso de una aplicación de minería de datos que nos brinda un entorno de computación en nube. En este trabajo, se propone un marco computacional de la nube, que se encarga de distribuir y programar una aplicación de minería de datos basados en clúster y su conjunto de datos. El objetivo principal de esta propuesta de marco para la programación de la Gran Minería de datos (FSBD) es disminuir el tiempo total de ejecución de la aplicación con la mínima

pérdida en la calidad de la minería. Se considera que la técnica de minería de datos basada en clústeres como una aplicación piloto para el marco.

Los resultados muestran un aumento de velocidad importante con una pérdida mínima de la calidad de la minería. Se obtuvo una relación de 2 de la normalizada actual makespan vis-a vis el makespan ideal. La calidad de la minería de escala bien con el número de grupos y el aumento del tamaño del conjunto de datos. Los resultados son prometedores, el fomento de la adopción del marco por los proveedores de la nube, de acuerdo a Ismail, Masud y Khan. [12].
Transplantation of Data Mining Algorithms to Cloud Computing Platform When Dealing Big Data

En este trabajo se hace una breve reseña de Computación en la nube y Big Data, se discutió la portabilidad de los algoritmos de minería de datos en general a la plataforma de computación en la nube. Se reveló que la plataforma de computación en la nube basada en Map-Reduce no puede resolver todos los problemas de grandes volúmenes de datos y minería de datos. El trasplante de los algoritmos de minería de datos generales al tiempo real plataforma de computación en la nube será una parte de la investigación se centra en computación en la nube y Big Data, según Wang y Zhao [13].

Big Data Processing in Cloud Computing Environments

Con el rápido crecimiento de las aplicaciones emergentes como el análisis social de redes, análisis Web semántica y análisis de redes de bioinformática, una variedad de datos a procesar sigue siendo testigo de un aumento rápido. La gestión eficaz y análisis de datos a gran escala supone un reto interesante pero crítico. Recientemente, los grandes datos han atraído mucha atención por parte de la academia, la industria, así como del gobierno. Este documento presenta varias técnicas de procesamiento de datos de grandes sistemas y aplicaciones aspectos. En primer lugar, desde el punto de vista de la gestión de datos de nube y los mecanismos de procesamiento de datos grandes, se presentan los temas clave de procesamiento de grandes volúmenes de datos, incluyendo la plataforma de computación en la nube, arquitectura de nube, y el esquema de base de datos de la nube de almacenamiento de datos. Siguiendo el marco de procesamientoMapReduce en paralelo, que a continuación se presentan estrategias MapReduce de optimización y aplicaciones reportadas en la literatura. Finalmente, se discuten las cuestiones abiertas y desafíos, y se exploran profundamente las direcciones de investigación en el futuro sobre el procesamiento de grandes volúmenes de datos en entornos de computación en nube, como dicen los autores del documento Big Data Processing in Cloud Computing Environments [14].

Método

Son necesarias las siguientes herramientas para proceder con la implementación:

AWS EC2

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad de cómputo con tamaño modificable en la nube. Está diseñado para facilitar a los desarrolladores la programación en la nube escalable basado en web.

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar la capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener y arrancar nuevas instancias de servidor en cuestión de minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, según cambien sus necesidades. Amazon EC2 cambia el modelo económico de la informática, ya que solo tendrá que pagar por la capacidad que realmente utilice. Amazon EC2 proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarse de los casos de error más comunes, así como se aprecia en la página oficial de Amazon [1].

El clúster fue lanzado en esta característica de AWS utilizando un nodo como master y otros dos nodos como esclavos.

Weka (versión 3.8.0)

Weka es un software que contiene un conjunto de algoritmos de aprendizaje automático para tareas de minería de datos. Los algoritmos o bien se pueden aplicar directamente a un conjunto de datos o llamadas de su propio código Java. Weka contiene herramientas para el procesamiento previo de datos, clasificación, regresión, clustering, reglas de asociación, y la visualización. También es muy adecuado para el desarrollo de nuevos sistemas de aprendizaje de máquina.

Weka es un software de código abierto publicado bajo la Licencia Pública General de GNU, así como se observa en la pagian web de la Universidad de Waikato [2].

Se utilizó la versión más reciente de Weka, la versión 3.8.0 disponible en su página web, esta contiene el algoritmo de minería de datos J48.

Starcluster(versión 0.91)

StarCluster es un conjunto de herramientas para implementación de clúster de código abierto para Elastic Compute Cloud (EC2) de Amazon publicado bajo la licencia LGPL.

StarCluster ha sido diseñado para automatizar y simplificar el proceso de construcción, configuración y gestión de grupos de máquinas virtuales en la nube EC2 de Amazon. StarCluster permite a cualquiera crear fácilmente un entorno de computación en la nube, adecuado para aplicaciones y sistemas informáticos distribuidos y paralelos, según el MIT [3].

Este software se utiliza para crear el clúster en las instancias de AWS, a través de un equipo local Ubuntu, se instaló la última versión disponible 0.91.

MpichCluster

MPI (MessagePassing Interface, Interfaz de Paso de Mensajes) es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.

MPICH se distribuye bajo una licencia BSD,Los pquetes binarios MPICH están disponibles en muchas distribuciones UNIX y para Windows. Por ejemplo, se encuentra con "yum" (en Fedora), "apto" (Debian / Ubuntu), "pkg_add" (FreeBSD) o "puerto" / "brew" (Mac OS), según el MPI [9].

Ya con las herramientas instaladas, se procede con la implementación:

Instalación de StarCluster

StarCluster es una recopilación de herramientas de código abierto para la computación en la nube, específicamente para ser utilizado con Amazon Elastic Compute Cloud (Amazon EC2).

StarCluster ha sido diseñado para automatizar y simplificar el proceso de creación, configuración y administración de clústers en máquinas virtuales de Amazon EC2. StarCluster permite crear fácilmente un entorno de computación para clúster en la nube. para aplicaciones y sistemas de computación paralela y distribuida.

StarCluster fue instalado en una maquina local con sistema operativo Ubuntu 14.04 LTS de 64 bits. Es necesario que dicho sistema tenga instalado el repositorio de software PyPI (Python PackageIndex), el cual permite la instalación de StarCluster.

Desde la terminal de Ubuntu se procede a instalar PyPI, como se observa en la imagen 2.

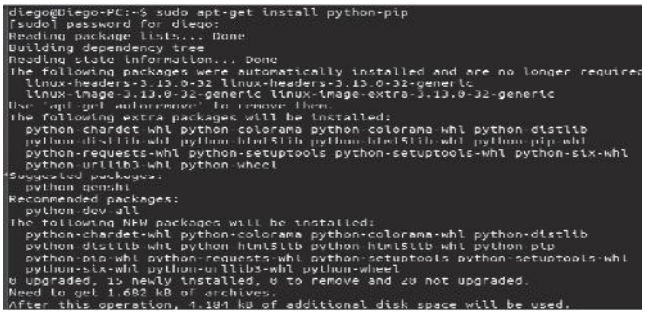


Imagen 2. Instalación de PyPi. Fuente: Autores.

Cuando se complete la instalación de PyPi, se procede con la instalación de StarCluster con el comando: *sudo easy_install StarCluster*, cuya ejecución se muestra en la imagen 3.

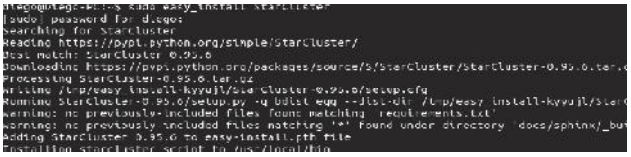


Imagen 3. Instalación de StarCluster. Fuente: Autores.

2. Configuración del clúster

El primer paso para configurar el clúster es crear y editar el archivo de configuración de StarCluster. Es importante utilizar las funciones en modo súper usuario de Ubuntu (comando sudo su) debido a que se necesitan utilizar ciertas funciones que no permite un usuario normal, el súper usuario y el usuario normal no tendrán ninguna relación de configuración o datos. Para crear el archivo de configuración de StarCluster se debe acceder a la ayuda a través del comando starclusterhelp, como se observa en la imagen 4.

```
diego@Diego-PC:~$ starcluster help
StarCluster - (http://star.mit.edu/cluster) (v. 0.95.6)
Software Tools for Academics and Researchers (STAR)
Please submit bug reports to starcluster@mit.edu

!!! ERROR - config file /home/diego/.starcluster/config does not

Options:
--------
[1] Show the StarCluster config template
[2] Write config template to /home/diego/.starcluster/config
[q] Quit

Please enter your selection:
```

Imagen 4. Comando de ayuda de StarCluster. Fuente: Autores.

Si es la primera vez que se ejecuta StarCluster aparecerá el mismo menú que el de la imagen 11, se debe seleccionar la opción 2, para que starcluster cree el archivo de configuración en su directorio (/home/user/.starcluster/config).

```
Please enter your selection: 2

>>> Config template written to /home/diego/.starcluster/con
>>> Please customize the config template
diego@Diego-PC:~$
```

Imagen 5. Creación del archivo de configuración de StarCluster. Fuente: Autores.

Se debe modificar el archivo de configuración de Starcluster con un editor de texto, en este caso se llevó a cabo con el editor gedit usando el comando gedit~/.starcluster/config, como se aprecia en la imagen 6.

```
root@diego-VirtualBox:/home/diego# gedit ~/.starcluster/config
```

Imagen 6. Ingreso al archivo de configuración de StarCluster. Fuente: Autores.

Dentro del archivo lo primero que se debe configurar son los valores de las claves de accesos que se obtuvieron en el apartado anterior, como se ve en la imagen 7.

```
[aws info]
AWS_ACCESS_KEY_ID = # Your Access Key ID here
AWS_SECRET_ACCESS_KEY = # Your Secret Access Key here
AWS_USER_ID = # Your 12-digit AWS Account ID here (no hyphens)
```

Imagen 7. Especificación de los campos de las claves de acceso en el archivo de configuración. Fuente: Autores.

Se modifican los valores correspondientes a los de códigos de seguridad de la cuenta, sin espacios, la clave de acceso de usuario (AWS USER ID) es de 12 números que están separados de a cuatro por guiones que también deben ingresarse.
En este archivo también deben configurarse otras características dentro de las cuales se encuentran el número de instancias a crear, el nombre del clúster, el usuario de administración del clúster, el tamaño de cada nodo y la clave de acceso del clúster.

Los signos de # son una etiqueta de comentario, por lo que no se tendrá en cuenta lo que esta después de este símbolo, cada símbolo es válido por línea, lo que quiere decir que la línea que no lo tenga no es un comentario.

Si se desea se pueden configurar otro valores, se puede elegir la región en la que se desplegara el clúster (AWS-REGION-NAME), configuraciones del proxy y claves extras (EC2_PRIVATE_KEY), como se ve en la imagen 8.

```
#########################################
## AWS Credentials and Connection Settings ##
#########################################
[aws info]
# This is the AWS credentials section (required).
# These settings apply to all clusters
# replace these with your AWS keys
AWS_ACCESS_KEY_ID = AKIAJI[redacted]
AWS_SECRET_ACCESS_KEY = ylbfxaxALcOwD[redacted]
# replace this with your account number
AWS_USER_ID= 8829-[redacted]-[redacted]
# Uncomment to specify a different Amazon AWS region  (OPTIONAL)
# (defaults to us-east-1 if not specified)
# NOTE: AMIs have to be migrated!
#AWS_REGION_NAME = eu-west-1
#AWS_REGION_HOST = ec2.eu-west-1.amazonaws.com
# Uncomment these settings when creating an instance-store (S3) AMI (OPTIONAL)
#EC2_CERT = /path/to/your/cert-asdf0as9df092039asdfi02089.pem
#EC2_PRIVATE_KEY = /path/to/your/pk-asdfasd890f200909.pem
# Uncomment these settings to use a proxy host when connecting to AWS
#AWS_PROXY = your.proxyhost.com
#AWS_PROXY_PORT = 8080
#AWS_PROXY_USER = yourproxyuser
#AWS_PROXY_PASS = yourproxypass
```

Imagen 8. Apartado de credenciales de seguridad en el archivo de configuración. Fuente: Autores.

La llave es un sistema de protección para acceder y modificar el clúster, posteriormente se creara dicha clave, se debe poner el nombre de la clave (keymykey, mykey es el nombre de la llave) se debe poner el directorio donde se creara dicha clave (KEY_LOCATION, en este caso ~/.starcluster/mykey.rsa), se puede agregar el número de llaves que se deseé, como en la imagen 9.

```
############################
## Defining EC2 Keypairs ##
############################
# Sections starting with "key" define your keypairs. See "starcluster createkey
# --help" for instructions on how to create a new keypair. Section name should
# match your key name e.g.:
[key mykey]
KEY_LOCATION=~/.starcluster/mykey.rsa

# You can of course have multiple keypair sections
# [key myotherkey]
# KEY_LOCATION=~/.ssh/myotherkey.rsa
```

Imagen 9. Apartado de definición de llaves en el archivo de configuración. Fuente: Autores.

En la parte de configuración del clúster se ingresa el nombre que se deseé al clúster (en este caso mycluster), el nombre de la llave definida anteriormente (KEYNAME = mykey), el número de nodos del clúster (CLUSTER_SIZE = 3), el usuario del clúster (CLUSTER_USER=sgeadmin), la imagen de la máquina de Amazon (Amazon Machine Images, AMI), en este caso se seleccionó ami-3393a45a el cual es el estándar de imagen de starcluster de Ubuntu 13.04 de 64 bits, que será el sistema operativo que se instalara en cada uno de los nodos, finalmente, se debe configurar el tamaño de la instancia, que para

evitar costos debe ser t1.micro (NODE_INSTANCE_TYPE=t1.micro)

```
[cluster mycluster]
# change this to the name of one of the keypair sections defined above
KEYNAME = mykey
# number of ec2 instances to launch
CLUSTER_SIZE = 3
# create the following user on the cluster
CLUSTER_USER = sgeadmin
# optionally specify shell (defaults to bash)
# (options: tcsh, zsh, csh, bash, ksh)
CLUSTER_SHELL = bash
# Uncomment to prepend the cluster tag to the dns name of all nodes created
# using this cluster config.  ie: mycluster-master and mycluster-node001
# If you choose to enable this option, it's recommended that you enable it in
# the DEFAULT_TEMPLATE so all nodes will automatically have the prefix
# DNS_PREFIX = True
# AMI to use for cluster nodes. These AMIs are for the us-east-1 region.
# Use the 'listpublic' command to list StarCluster AMIs in other regions
# The base i386 StarCluster AMI is ami-9bf9c9f2
# The base x86_64 StarCluster AMI is ami-3393a45a
# The base HVM StarCluster AMI is ami-6b211202
NODE_IMAGE_ID = ami-3393a45a
# instance type for all cluster nodes
# (options: m3.large, c3.8xlarge, i2.8xlarge, t2.micro, hs1.8xlarge, c1.xlarge
c3.2xlarge, m2.xlarge, m2.2xlarge, t2.small, r3.2xlarge, t1.micro, cr1.8xlarge
i2.xlarge, m3.medium, cc2.8xlarge, m1.large, cg1.4xlarge, i2.2xlarge, c3.large
m2.4xlarge, m1.xlarge, m3.xlarge)
NODE_INSTANCE_TYPE = t1.micro
# Launch cluster in a VPC subnet (OPTIONAL)
```

Imagen 10. Apartado de opciones del clúster en el archivo de configuración. Fuente: Autores.

Como se explicó anteriormente la llave (keypair), sirve para darle permisos al StarCluster instalado en la maquina local para poder configurar, lanzar y editar el clúster en Amazon, es por eso que el primer paso para lanzar el clúster consiste en crearla(s), con los datos que se ingresaron en el apartado de definición de llaves (imagen 16), como todos los pasos debe hacerse con el súper usuario de Ubuntu, con el siguiente comando Starclustercreatekeymykey –o ~/.starcluster/mykey.rsa, mykey el nombre de la clave, -o porque está en un archive de salida y la dirección donde se creara la llave. Se procede como se ve en la imagen 11.

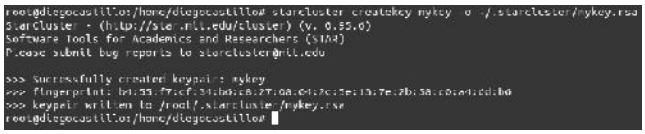


Imagen 11. Creación de la llave. Fuente: Autores.

En la carpeta indicada se puede visualizar el archivo de configuración y la llave, como en la imagen 12.

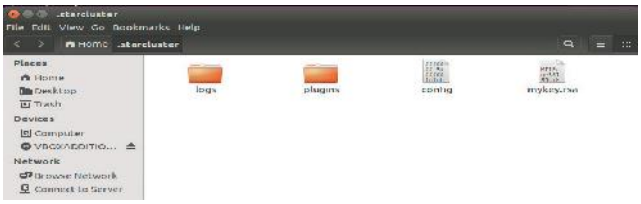


Imagen 12. Carpeta .starcluster con el archivo de configuración y la llave. Fuente: Autores.

Luego de tener todas las configuraciones, la llave creada y haber ingresado como súper usuario, se procede a inicializar el clúster con el comando:

Starcluster start mycluster, como se ve en la imagen 13.

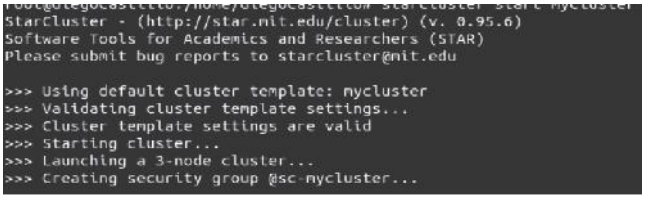


Imagen 13. Inicio del clúster. Fuente: Autores.

Luego de que StarCluster termine de configurar el clúster y sus instancias se debe acceder a la consola de administración de AWS, como en la imagen 14.

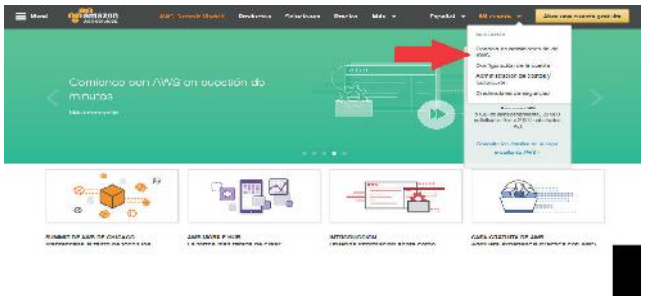


Imagen 14. Accediendo a la consola de administración de AWS. Fuente: Autores.

Se da click en instancias ,en el menú lateral se podrá ver la lista de nodos creados, aquí se puede comprobar que el clúster junto con los nodos deseados se crea correctamente. Se puede verificar además el tamaño del nodo y el sistema operativo instalado.

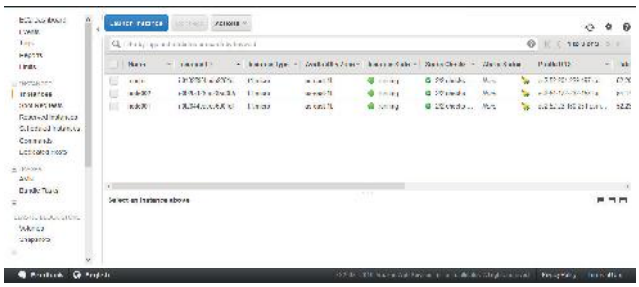


Imagen 15. Lista de instancias creadas en AWS. Fuente: Autores.

Se debe parar el clúster cada vez que deje de usarse con el comando:

Starcluster stop mycluster
Para iniciar el clúster luego de pararlo se utiliza
Starcluster start -x mycluster

Para terminarlo (borrarlo) se utiliza el comando:
Starclusterterminatedmycluster

3. Probando el clúster y sus nodos

SSH (Secure Shell) es el intérprete de órdenes que sirve para acceder a maquinas remotas a través de una red, permite manejar por completo otro computador por medio de un intérprete de comandos y también puede redirigir tráfico de X (Sistema de Ventanas X) para poder ejecutar programas gráficos si tenemos ejecutando un Servidor X (en sistemas Unix y Windows).

Además de la conexión a otros dispositivos, SSH permite copiar datos de forma segura (tanto archivos sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

Se utilizara SSH para comprobar el correcto funcionamiento del clúster y posteriormente se configurara correctamente para poder acceder desde el nodo maestro a cada uno de los nodos.

Se debe dar permisos de acceso a la llave para poder probar los nodos, los permisos son para que la llave se pueda leer por el propietario (chmod 400), como se ve en la imagen 16.

```
root@diegocastillo:/home/diegocastillo# chmod 400 ~/.starcluster/mykey.rsa
```

Imagen 16. Cambiando permisos de acceso. Fuente: Autores.

Para comprobar el funcionamiento del clúster se usa el comando de conexión de ssh:
ssh –i ~/.starcluster/mykey.rsaubuntu@(dirección del nodo) como en la imagen 17.

Imagen 17. Comando SSH para conexión al nodo maestro. Fuente: Autores.

Imagen 18. Conección en el nodo maestro. Fuente: Autores.

Para obtener la dirección de los nodos debe accederse a la lista de nodos, la columna correspondiente a DNS público (Public DNS), esta es la dirección que debe utilizarse para conectarse a un nodo a través de SSH.

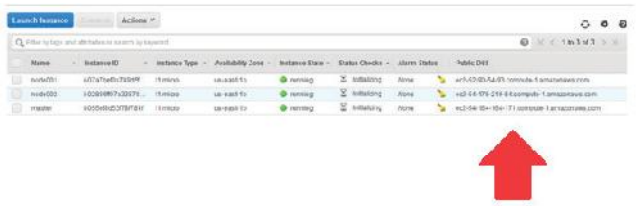


Imagen 19. Dirección de cada nodo. Fuente: Autores.

Con el comando qhost se puede verificar la capacidad del clúster

```
root@master:~# qhost
HOSTNAME                ARCH         NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------
global                  -               -     -       -       -       -       -
master                  linux-x64       1  0.01  589.6M   98.9M     0.0     0.0
node001                 linux-x64       1  0.01  589.6M   80.3M     0.0     0.0
node002                 linux-x64       1  0.01  589.6M   80.8M     0.0     0.0
```

Imagen 20. Capacidad del cluster. Fuente: Autores.

4. Instalación de SSH en el clúster

El SSH permitirá conectarse desde el nodo maestro a los nodos esclavos y viceversa, por lo cual debe generarse una clave que conozcan los 3 nodos del clúster para permitir el acceso entre ellos.
Con la sesión de súper usuario se debe acceder al nodo maestro a través del comando de starcluster: starclustersshmastermycluster

En el nodo maestro debe generarse la clave que conocerán los tres nodos y que permitirá la conexión entre ellos, el comando que sirve, como se ve en la imagen 21.

```
root@master:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): keyp.pem
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in keyp.pem.
Your public key has been saved in keyp.pem.pub.
The key fingerprint is:
1c:01:95:c0:19:b6:00:e0:d5:34:87:77:a9:38:bc:2d root@master
The key's randomart image is:
+--[ RSA 2048]----+
|...o++*Bo..      |
|. .   ==o.+      |
| .  . + +        |
|     + o .       |
|      + S        |
|     E .         |
|      .          |
|                 |
|                 |
+-----------------+
```

Imagen 21. Creación de la llave en el nodo maestro para la conexión entre los nodos. Fuente: Autores.

Desde el usuario root en el nodo maestro (comando sudo su), debe copiarse la clave creada a los nodos esclavos con el comando:

ssh-copy-id –i ~/.ssh/id_rsa.pub root@(dirección del nodo), esto se repite para los dos nodos esclavos.

Imagen 22. Copia de la llave del nodo maestro a los nodos esclavos. Fuente: Autores.

Además evita que se tenga que poner la dirección de la clave cada vez que quiera accederse a los nodos.

Se accede a los nodos esclavos con el comando ssh, usuario root y dirección de los nodos.

Imagen 23. Comando para acceder desde el nodo maestro a los nodos esclavos. Fuente: Autores.

Se puede visualizar que se ingresa desde el nodo maestro al nodo esclavo por la dirección que aparece.

Imagen 24. Visualización del acceso desde el nodo maestro al nodo esclavo 2. Fuente: Autores.

5. Ejecución del programa en el clúster y Montaje del software en el clúster

Para subir el software que se quiere ejecutar en el clúster se debe ubicar la carpeta que contiene el proyecto ejecutable que netbeans o eclipse generan (Archivo jar, librerías, Dataset).
Se debe tener en cuenta la carpeta donde se ubica el proyecto ejecutable.

Imagen 25. Proyecto dist de archivos generados por Netbeans. Fuente: Autores.

Con el clúster ejecutando se procede a subir dicho archivo a todos los nodos del mismo.

Se utiliza el comando put de starcluster para cargar la carpeta indicándole primero la dirección del archivo localmente y luego la dirección donde se desea poner la carpeta en el nodo, se debe indicar el nodo al que se desea subir, como en la imagen 26. En caso de no hacerlo el proyecto será subido al nodo maestro.

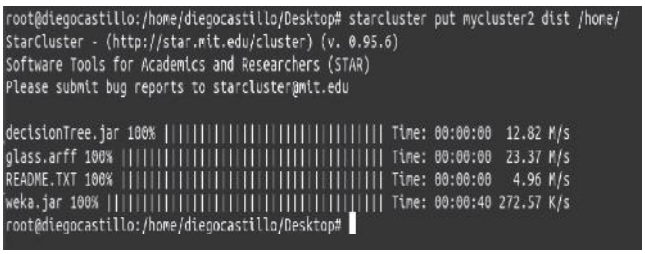


Imagen 26.Carga del proyecto al nodo maestro del clúster. Fuente: Autores.

Con el comando $ starcluster put mycluster2 --node node001 dist /home/ se procede a cargar la carpeta a los nodos.

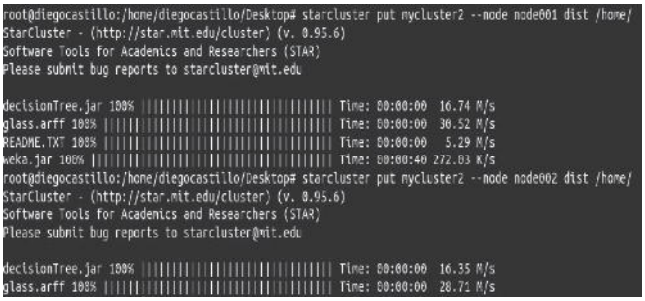


Imagen 27. Carga del proyecto a los nodos esclavos del clúster. Fuente: Autores.

Es importante verificar que se halla copiado correctamente el proyecto a los nodos, primero entrando con ssh en cada nodo y se ubica en la carpeta home del nodo utilizando el comando ls, para que imprima los archivos disponibles en dicha carpeta.

6. Ejecución del software en el clúster

Lo primero que se debe hacer en el clúster es habilitar el plugin de mpich2 en el archivo de configuración del clúster, como en la imagen 28.

Imagen 28. Configuración del plugin mpich2 en el archivo de configuración de starcluster. Fuente: Autores.

Después de reiniciar el clúster se debe correr el plugin mpich2, como en la imagen 29.

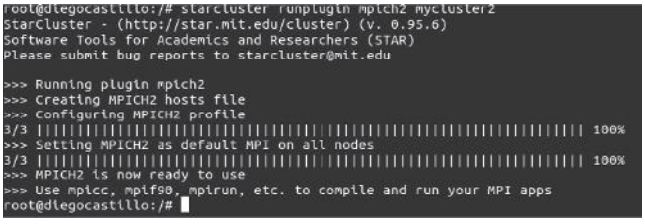


Imagen 29. Ejecucuión del plugin de mpich2. Fuente: Autores.

Para ejecutar el programa se debe utilizar el comando de mpimpirun de java para poder ejecutar en todos los nodos el ejecutable JAR en todos los nodos, como en la imagen 30.

```
README.TXT  decisionTree.jar  dist  glass.arff  lib
root@master:/home/dist# mpirun java -jar decisionTree.jar
```

Imagen 30. Subiendo el proyecto al clúster. Fuente: Autores.

7. Validación del código

El código que utiliza el algoritmo fue tomado de un proyecto propiedad de Madipalli, Gangsetty, Gangdhara y Malla público en un repositorio de Github [18], pero además se encuentran variaciones a través de internet y es posible encontrar la documentación de dichos métodos y clases en la página de la Universidad de Waikato [17]

Primero se procede a cargar el Dataset que debe ser de extensión arff para evitar problemas con otros tipos de archivos, la clase BufferedReader de java permite cargar dicho Dataset a un objeto, como en la imagen 31.

```
BufferedReader datafile = readDataFile("glass.arff");
```

Imagen 31. Carga del Dataset al objeto datafile. Fuente: Autores.

Se crea un objeto tipo Instances que será el que contendrá el objeto datafile para manejar el Dataset con los métodos de Weka. Se debe además ubicar ese objeto como el último atributo de la clase para que funcione, como en la imagen 32.

```
Instances data = new Instances(datafile);
data.setClassIndex(data.numAttributes() - 1);
```

Imagen 32. Creación objeto Instances y haciéndolo el último atributo de la clase. Fuente: Autores.

Para ejecutar la validación cruzada tantas veces como se desee, se pasa el parámetro data y el número de iteraciones deseadas (10) al método de validación cruzada. En un objeto de array bidimensional, como en la imagen 33.

```
// Do 10-split cross validation
Instances[][] split = crossValidationSplit(data, 10);
```

Imagen 33. iteraciones y paso de parámetro data al método de validación cruzada. Fuente: Autores.

El método de validación cruzada se encarga de separar los resultados del algoritmo en datos de prueba y datos de entrenamiento durante todas las iteraciones como en la imagen 34.

```
public static Instances[][] crossValidationSplit(Instances data, int numberOfFolds) {
    Instances[][] split = new Instances[2][numberOfFolds];

    for (int i = 0; i < numberOfFolds; i++) {
        split[0][i] = data.trainCV(numberOfFolds, i);
        split[1][i] = data.testCV(numberOfFolds, i);
    }

    return split;
}
```

Imagen 34. Método de validación cruzada. Fuente: Autores.

Se debe usar un conjunto de clasificadores para la ejecución del algoritmo con validación cruzada, estos están contenidos dentro de un clasificador llamado modelo el cual tiene cuatro atributos el árbol de decisión J48, parte, tabla de decisión, y árbol de decisión de un nivel, como en la imagen 35.

```
// Use a set of classifiers
Classifier[] models = {
        new J48(), // a decision tree
        new PART(),
        new DecisionTable(),//decision table majority classifier
        new DecisionStump() //one-level decision tree
};
```

Imagen 35. Clasificador modelo. Fuente: Autores.

Para cada nivel del modelo se debe correr un ciclo que permita coleccionar cada grupo de predicciones para el modelo actual con el vector rápido, además, se deben probar los datos de prueba y construir el modelo con los datos de entrenamiento. Finalmente se calcula la precisión global del algoritmo en todas las divisiones, como en la imagen 36.

```
// Run for each model
for (int j = 0; j < models.length; j++) {

    // Collect every group of predictions for current model in a FastVector
    FastVector predictions = new FastVector();

    // For each training-testing split pair, train and test the classifier
    for (int i = 0; i < trainingSplits.length; i++) {
        Evaluation validation = classify(models[j], trainingSplits[i], testingSplits[i]);

        predictions.appendElements(validation.predictions());

        // Uncomment to see the summary for each training-testing pair.
        System.out.println(models[j].toString());
        System.out.println(validation.toClassDetailsString());
        System.out.println(validation.toSummaryString()
        );
        System.in.read();
    }

    // Calculate overall accuracy of current classifier on all splits
    double accuracy = calculateAccuracy(predictions);

    System.out.println("Accuracy of " + models[j].getClass().getSimpleName() + ": "
            + String.format("%.2f%%", accuracy)
            + "\n                                    " + models[j].toString() );
}
```

Imagen 36. Ciclo que se corre en todos los niveles que permite predecir, crear y probar el árbol y calcular la precisión global del algoritmo. Fuente: Autores.

Resultados

1. Resultados del algoritmo

En la evaluación cruzada se realizan tantas evaluaciones como se indica en el parámetro de iteraciones folds. Se dividen las instancias en tantas carpetas como indica este parámetro y en cada evaluación se toma una carpeta como dato de prueba

para que el resto sean datos de entrenamiento que permitan la construcción del modelo (el árbol), los errores calculados son el promedio de todas las ejecuciones.

En promedio se dividieron los datos en grupos de 21 de 214 instancias que posee el Dataset glass que posteriormente se clasificaron con el parámetro J48 y según la posición se verifica el éxito o error.

En la gráfica 1 se presenta el porcentaje de éxito y error en una prueba con 10 iteraciones del parámetro folds.

Grafica 1. Porcentaje de éxito y error por iteración. Fuente: Autores.

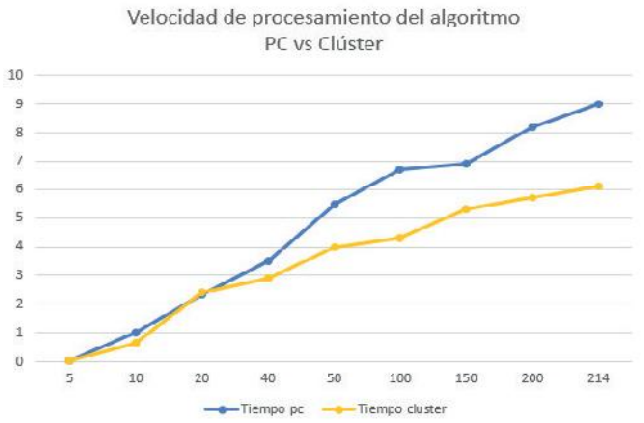
Se observa que aunque el porcentaje de éxito de los datos de prueba de vidrios evaluados en el algoritmo es mayor que el de errores, el algoritmo no mejora con el número de iteraciones hechas, pero se puede establecer la exactitud que tiene el algoritmo con estos datos es en promedio 67,976%.

Resultados procesamiento en clúster

Una de las ventajas de procesar el algoritmo en el clúster es la velocidad de ejecución de éste, debido a que el mismo algoritmo que se ejecuta en una sola maquina es ejecutado en todos los nodos del clúster mejorando el rendimiento notablemente; muchas veces la cantidad de datos es tan extensa que los computadores normales no pueden ejecutar procesos de minería de datos, por lo que es en estos casos que se requiere la combinación de tecnologías como computación a través de clúster en la nube.

El número de iteraciones hace que la ejecución del algoritmo crezca exponencialmente debido al número de procesos que se deben hacer al evaluar los datos de prueba y construir el modelo con los datos de entrenamiento.

La gráfica 2 muestra el tiempo de ejecución del algoritmo J48 evaluado en número de iteraciones de validación cruzada contra segundos.



Grafica 2. Velocidad de procesamiento en segundos PC vs Clúster. Fuente: Autores.

Se puede visualizar que la diferencia de tiempo de ejecución es notoria desde las 40 iteraciones de validación cruzada, es evidente como en el computador de escritorio el tiempo de ejecución crece exponencialmente, para que dicho proceso ocurra en el clúster se debe utilizar una base de datos muy extensa.

Conclusiones

Como se evidencia en los antecedentes, la computación en la nube para la ejecución de minería de datos es uno de los temas más recurrentes en los últimos años, ya que la posibilidad de poder analizar cantidades gigantescas de datos para evaluar posibles soluciones, nuevos productos, tendencias y demás es de suma importancia para las corporaciones y gobiernos; con aplicaciones desde marketing hasta misiones espaciales, las dos tecnologías juntas no tienen límites.

El presente artículo demuestra que dicho proceso se puede llevar a cabo con software libre, como Weka y Starcluster utilizando la capa gratuita de los servicios de Amazon Web Service. Si se desea cantidades gigantes de datos estas herramientas tienen soporte escalable que facilitaran dicho proceso.

El algoritmo J48 permite establecer características que agrupan objetos según ciertos criterios. En este artículo se demuestra cómo dicho algoritmo, junto con el proceso de validación cruzada, crean modelos que son probados con todos los datos disponibles que establecen el porcentaje de aciertos y errores de dicho algoritmo. Dichos resultados sirven para determinar qué tanto debe acoplarse el proceso para obtener mejores resultados.

La ejecución de procesos en el clúster muestra mejores resultados con respecto a computadores domésticos a medida que la cantidad de datos crece y el procesamiento supera la capacidad de estos ordenadores. Además, el número de instancias que se implementen en un clúster pueden llegar a ser contraproducentes debido a que pueden no necesitarse. Esto se debe a que no se ha cumplido con la capacidad de las instancias existentes. Por eso es recomendable utilizar tecnologías escalables que permitan administrar el número de nodos y tamaño de los mismos, dichas especificaciones son propuestas por Amazon Web Service en su página web.

El Dataset utilizado contiene 251 instancias con 11 campos, es un archivo de tamaño pequeño (1mb) que no utiliza el clúster en toda su capacidad, se estima que un tamaño aceptable para el software y las instancias utilizadas (t1.miro) es de aproximadamente 100 mbs máximo, para evitar costos por procesamiento y almacenamiento.

De los autores

Jairo Jamith Palacios Rozo: Ingeniero de Sistemas Universidad Distrital Francisco José de Caldas – Colombia. Profesor de planta Universidad Colegio Mayor de Cundinamarca. Especialización Universidad Antonio Nariño Especialización En Administración de Empresas. Magister Universidad Santo Tomás Maestria en Educación. jjpalacios@unicolmayor.edu.co

Referencias

[1]. Amazon. (2016). ec2 instances. 22 de junio de 2016, de Amazon sitio web: https://aws.amazon.com/es/ec2/

[2]. Universidad de Waikato. (2010). Weka. 26 de junio de 2016, de Universidad de Waikato sitio web: http://www.cs.waikato.ac.nz/ml/weka/

[3]. Instituto de tecnología de Massachusetts (MIT). (2016). starcluster. 19 de junio de 2016, de MIT sitio web: http://star.mit.edu/cluster/index.html

[4]. Liliana Hernández Cervantes, Alfredo j. Santillán González, Reyna e. caballero cruz. (2004). Clúster. Revista digital universitaria, 4, 1,2.

[5]. Maestría de exploración de datos de datos y descubrimiento del conocimiento. (2015) ¿qué es data mining? 13 de junio de 2016, de Universidad de Buenos Aires, UBA, Argentina, sitio web: http://datamining.dc.uba.ar/datamining/index.php/que-es-data-mining

[6]. Ingrid Wilford Rivera, Alejandro Rosete Suárez. Alfredo Rodríguez Díaz. . (2010). Aplicación de la minería de datos para el análisis de información clínica. estudio experimental en cardiopatías isquémicas. 10 de junio de 2016, de revista cubana de información médica sitio web: http://www.rcim.sld.cu/revista_18/articulos_htm/mineriadatos.htm.

[7]. Microsoft. (2016). Validación cruzada (Analysis Services - Minería de datos). 2 de julio de 2016, de Microsoft Sitio web: https://msdn.microsoft.com/es-es/library/bb895174.aspx

[8]. García Jiménez, María - Álvarez Sierra, Aránzazu. (2010). Análisis de Datos en WEKA – Pruebas de Selectividad. 29 de junio de 2016, de Universidad Carlos III, Madrid, España

[9]. MPI. (2000). MPI. 10 de junio de 2016, de Ubuntu Sitio web: http://www.mpich.org/downloads/

[10]. D. Chiu and G. Agrawal, "Evaluating caching and storage options on the Amazon Web Services Cloud," 2010 11th IEEE/ACM International Conference on Grid Computing, Brussels, 2010, pp. 17-24. doi: 10.1109/GRID.2010.5697949 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5697949&isnumber=5697799

[11]. R. Gupta, Moinuddin and P. Kumar, "Cloud computing data mining to SCADA for energy management," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-6. doi:

10.1109/INDICON.2015.7443687 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7443687&isnumber=7443105

[12]. L. Ismail, M. M. Masud and L. Khan, "FSBD: A Framework for Scheduling of Big Data Mining in Cloud Computing," 2014 IEEE International Congress on Big Data, Anchorage, AK, 2014, pp. 514-521. doi: 10.1109/BigData.Congress.2014.81 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6906823&isnumber=6906742

[13]. Y. Wang and Y. W. Zhao, "Transplantation of Data Mining Algorithms to Cloud Computing Platform When Dealing Big Data," Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2014 International Conference on, Shanghai, 2014, pp. 175-178. doi: 10.1109/CyberC.2014.39 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6984302&isnumber=6984259

[14]. C. Ji, Y. Li, W. Qiu, U. Awada and K. Li, "Big Data Processing in Cloud Computing Environments," 2012 12th International Symposium on Pervasive Systems, Algorithms and Networks, San Marcos, TX, 2012, pp. 17-23. doi: 10.1109/I-SPAN.2012.9 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6428800&isnumber=6428797

[15]. UCI. (2006). GlassIdentification Data Set . 8 de junio de 2016, de UCI, tabla basada en Atribute information, Sitio web: https://archive.ics.uci.edu/ml/datasets/Glass+Identification

[16]. Hernández Cervantes, Liliana; Santillán González, Alfredo; Caballero Cruz,Reyna . (2004). MAESTROS Y ESCLAVOS. UNA APROXIMACIÓN A LOS CÚMULOS DE COMPUTADORAS. Revista Digital Universitaria, UNAM, 4, 4.Disponible http://www.revista.unam.mx/vol.4/num2/art3/jun_art3.pdf

[17]. Weka. (2010). Use WEKA in your Java code. 1 de septiembre de 2016, de TheUniversity of Waikato Sitio web: https://weka.wikispaces.com/Use+WEKA+in+your+Java+code

[18]. S. Madipalli, A. Gangisetty, V.Gangdhara, K. Malla. Codigo java utilizado, gangisettiarjun. (2015). Predicting Cardiac and Diabetic Problems using EHR research data. 1 de octubre de 2016, de github Sitio web: https://github.com/gangisettiarjun/HealthCare_PredictiveAnalytics