ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"WikiGrep Distribuido: Búsquedas avanzadas en la Wikipedia"

TRABAJO DE GRADUACIÓN

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Presentado por:

Irene Carolina Varas Palomeque,

Gabriel Antonio Paladines Herrera

Guayaquil – Ecuador

2009

AGRADECIMIENTO

A Dios nuestro señor, a quien debemos agradecer todo en esta vida.

A nuestros padres, nuestros hermanos, familiares y amigos por su apoyo y palabras de aliento en los momentos difíciles.

A nuestra maestra y amiga, Ing. Cristina Abad por sus consejos, enseñanzas y guía en la realización de esta tesis.

DEDICATORIA

A Dios, a nuestros padres, hermanos, familiares y amigos por toda la fe y esperanza depositada en nosotros.

TRIBUNAL DE GRADUACIÓN

Ing. Cristina Abad

DIRECTOR DEL PROYECTO

Ing. Fabricio Echeverría

EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**".

Irene Carolina Varas Palomeque

Gabriel Antonio Paladines Herrera

RESUMEN

En este proyecto se ha elaborado un motor de búsqueda que soporta expresiones regulares y cuyo repositorio de datos es la Wikipedia la enciclopedia libre, el sistema permite el ingreso de una expresión regular y por medio de un requerimiento asíncrono inicializa un clúster EC2, hace el grep dentro de todos los documentos y cuando obtiene la respuesta muestra a manera de lista los resultados, cada línea contiene el patrón encontrado y un enlace a la página de la Wikipedia del artículo.

En el desarrollo de este proyecto se hace uso de los servicios de Amazon, de librerías desarrolladas en java para la manipulación de páginas de la Wikipedia, Hadoop framework y los datasets previamente cargados en Amazon.

Se realizaron pruebas de búsquedas con varias expresiones regulares, estas búsquedas no fueron posibles de realizar en los motores de búsqueda tradicionales, ni en el motor de búsqueda de la propia Wikipedia, puesto que las expresiones regulares buscan texto que siga un patrón y no un texto específico.

Las pruebas realizadas muestran que un sistema de búsquedas avanzadas puede ser implementado con un bajo costo y alta escalabilidad utilizando servicios de cloud computing y procesamiento masivo de datos.

ÍNDICE GENERAL

ABREVIATURAS

JSP	Java servlet pages
XML	Extensible Markup Language
EC2	Elastic Computing Cloud
S3	Simple Storage Service
EMR	Elastic MapReducce
AMI	Amazon Machine Image
HDFS	Hadoop File System

ÍNDICE DE FIGURAS

FIGURA DESCRIPCIÓN PÁG

ÍNDICE DE TABLAS

INTRODUCCIÓN

La Wikipedia[1] es la enciclopedia en línea donde todos pueden contribuir. Está considerada en el top ten de los sitios más visitados en la Internet de acuerdo a Alexa.com.[2] Desde sus inicios en el 2001 ha crecido de manera exponencial, y ahora incluye acerca de cuatro millones de páginas [1]. Sus artículos pueden ser descargados en formato XML, para uso personal o con fines educativos. Todo el contenido de las páginas tiene licencias múltiples que permiten que su contenido sea usado y redistribuido, como Creative Commos Atribution-ShareAlike 3.0 Licence y GNU Free Documentation Licence (GFDL) [2]. Debido a la gran cantidad de información que contienen la Wikipedia, el procesarla en un solo computador puede llegar a ser ineficiente. Más aún, su alta tasa de crecimiento hace que la situación empeore con el paso del tiempo. Por esta razón, es necesario un procesamiento distribuido de los datos de la misma, el cual permita obtener resultados en poco tiempo, de manera escalable y eficiente.

Los objetivos:

- Utilizar técnicas de procesamiento masivo de datos para realizar búsquedas avanzadas de texto dentro de documentos.

- Utilizar expresiones regulares para la búsqueda de texto dentro de documentos y de esta manera mostrar resultados más precisos.

---

1 http://www.wikipedia.org

2 http://www.alexa.com

- Diseñar una interfaz Web que permita ingresar búsquedas basadas en expresiones regulares, y muestre los resultados obtenidos del procesamiento masivo de la misma, de una manera entendible al usuario.

Nuestra motivación:

Editores de texto y otras herramientas normalmente proveen búsquedas basadas en expresiones regulares (por ejemplo, el comando `grep` en sistemas *NIX). Para este tipo de búsquedas no se puede generar índices que mejoren su rendimiento, ya que deben trabajar con el texto completo del corpus a ser analizado.

El problema surge cuando se trata de buscar sobre una cantidad muy grande de documentos y la capacidad de una sola máquina no es suficiente. La primera máquina de búsqueda para la Web, “The Archie directory service” fue creada en el año 1992, esta utilizaba expresiones regulares para obtener el nombre de los archivos que coincidían con el patrón de búsqueda [3]. El crecimiento de la información en la Web originó que las máquinas de búsqueda utilicen otras técnicas, como crear índices con información relevante de cada página, disminuyendo los tiempos de respuesta, pero limitando la exactitud de la búsqueda.

La Wikipedia es una enciclopedia libre con una gran cantidad de contenido disponible, que ha puesto a disposición sus artículos en formato XML que llegan a pesar hasta 1TB. En la actualidad las búsquedas dentro del contenido de la Wikipedia se realizan por el uso de índices, comodines y utilizando expresiones booleanas, y están restringidas al uso de expresiones regulares [4]. Este proyecto busca utilizar el paradigma MapReduce como técnica de procesamiento distribuido para procesar grandes cantidades de datos en una búsqueda basada en expresiones regulares.

Capítulo 1

1. Análisis de los datasets de la Wikipedia

1.1 Introducción

Los datos serán provistos en un solo archivo XML que contiene todos los artículos. Un artículo es representado como la entidad página. Los datos a utilizar se obtendrán desde "English Wikipedia dumps in XML: http://download.wikimedia.org/enwiki/".

1.2 Formato de los datos

La entidad página contendrá información como: título, restricciones y revisiones.

Para llevar un completo historial del contenido de los artículos, la entidad página contiene una o más revisiones <revision> ... </revision>. Las revisiones poseen información como: fecha de la revisión, información del usuario que contribuye a la revisión, los comentarios y el texto a mostrar.

El HTML de los artículos se encuentra dentro de la entidad <text> .. </text>, donde se utiliza los caracteres de escape <foo>.

Los links dentro de la Wikipedia son representados por [[destination]]. Si el nombre de la página de destino no está destinado a ser el enlace de texto, entonces se representará por: [[destination_page|display_text]].

1.3 Un ejemplo del archivo XML

```
- <mediawiki xml:lang="en">
  - <page>
      <title>Page title</title>
      <restrictions>edit=sysop:move=sysop</restrictions>
    - <revision>
        <timestamp>2001-01-15T13:15:00Z</timestamp>
      - <contributor>
          <username>Foobar</username>
        </contributor>
        <comment>I have just one thing to say!</comment>
        <text>A bunch of [[text]] here.</text>
        <minor/>
      </revision>
    - <revision>
        <timestamp>2001-01-15T13:10:27Z</timestamp>
      - <contributor>
          <ip>10.0.0.2</ip>
        </contributor>
        <comment>new!</comment>
        <text>An earlier [[revision]].</text>
      </revision>
    </page>
  - <page>
      <title>Talk:Page title</title>
    - <revision>
        <timestamp>2001-01-15T14:03:00Z</timestamp>
      - <contributor>
          <ip>10.0.0.2</ip>
        </contributor>
        <comment>hey</comment>
        <text>WHY YOU LOCK PAGE??!!!</text>
      </revision>
    </page>
  </mediawiki>
```

Figura 1. Formato XML artículo de la Wikipedia

1.4 Procesando el dataset

Para procesar los dataset hemos hallado dos alternativas; una descrita en un seminario dictado por Cloudera “Wikipedia Datasets for the Hadoop Hack” [5]; en el que describren que hacen un proceso de "flattening", este proceso consiste en dividir el documento XML en numerosos archivos más pequeños que corresponden aproximadamente al tamaño de la tarea del Map. Cada artículo será representado como un simple registro dentro del Mapper(Text), es decir, el proceso pone toda la información de un artículo en una línea del archivo, dado que de forma nativa hadoop lee los archivos y los distribuye línea a línea entre los mappers.

La otra alternativa es usar la librería MapReduce para Hadoop Cloud9, que describiremos con más detalle a continuación.

1.5 Cloud9, Librería MapReduce para Hadoop

Cloud9 fue diseñada como herramienta para la enseñanza y para dar soporte a la investigación en el procesamiento de texto. Fue utilizada en un seminario de "cloud computing" en la Universidad de Maryland. Al igual que Hadoop, Cloud9 es distribuida bajo Apache License. [6]

Cloud9 provee clases para trabajar con los XML dumps de la Wikipedia.

La siguiente tabla muestra las clases disponibles:

Clases Disponibles	
NumberWikipediaArticles	Este programa construye el "mapping" a partir de los títulos de los artículos de la Wikipedia (docids) a enteros numerados secuencialmente (docnos).
WikipediaDocnoMapping	Objeto que mapea entre Wikipedia docids (títulos de artículos) a docnos (enteros numerados secuencialmente).
WikipediaPage	Objeto que representa una página de la Wikipedia.

WikipediaPageInputFormat	Hadoop InputFormat para procesar páginas de la Wikipedia a partir de XML dumps.
WikipediaPageInputFormat.WikipediaPageRecordReader	Hadoop RecordReader para leer páginas de la Wikipedia a partir de XML dumps.
WikipediaPagesBz2InputStream	Clases para trabajar con archivos de los artículos de la Wikipedia comprimidos a bz2.

Tabla I .- Clases disponibles Cloud[9].

Realizando un análisis de las opciones para trabajar con los datasets de la Wikipedia tales como; procesar los datasets antes de utilizarlos o utilizar la librería Cloud9, nos decidimos por la segunda opción.

	Flattening	**Cloud[9]**
Trabaja directamente con los datasets de la wikipedia	No	Si
Utiliza hadoop de manera nativa	Si	No
Escalabilidad	No	Si

Tabla II.- Comparativo Proceso Flattening vs. Cloud[9].

Capítulo 2

2. Diseño e Implementación de la solución

2.1 Introducción

En este capítulo detalla cada uno de los elementos y fases utilizadas para el diseño e implementación de la solución.

2.2 Análisis de Alternativas

Considerando que en la ESPOL no existe un clúster de computadores para el procesamiento distribuido con Hadoop, se utilizó los servicios de Amazon (Amazon Web Services) para levantar clusters computacionales bajo demanda. Para este proyecto se consideró las dos alternativas descritas a continuación.

2.2.1 Amazon EC2 , Hadoop y Amazon S3

Levantar un clúster de computadoras utilizando el servicio EC2 (Elastic Computing Cloud) con Hadoop instalado, de "*n*" nodos y utilizar el servicio S3 (Simple Storage Service) para el almacenamiento de los datos.

Bajo este esquema, el clúster puede ser levantado y configurado para cada consulta, o puede mantenerse activo durante varias horas (por ejemplo, durante las 8 horas de una jornada laboral típica) y servir así a todas las solicitudes que se realicen durante ese periodo.

2.2.2 Elastic Map Reduce

Amazon Elastic MapReduce (EMR) incrusta automáticamente una implementación de Hadoop en las instancias de Amazon EC2, sub-dividiendo los datos de un flujo de trabajo en pequeñas partes, de forma que ellos puedan ser procesados (la función "map") en paralelo y, eventualmente, recombinado los datos en una solución final (la función "reduce"). Amazon S3 sirve como fuente para los datos de entrada, así también como el destino para el resultado final [7].

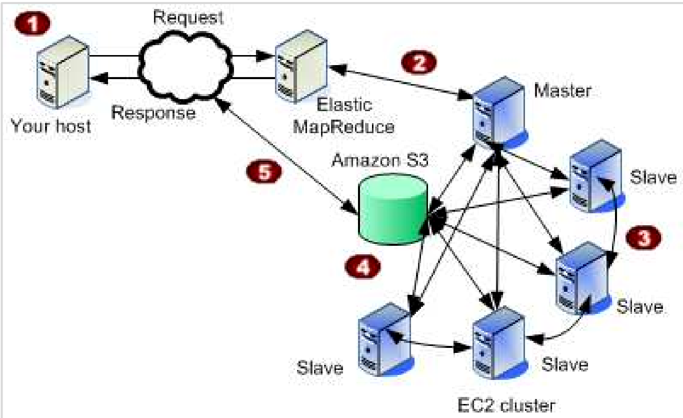

Figura 2. Arquitectura Elastic Mapreduce

1	Cargar los datos en Amazon S3, así como también el mapper y el reducer que procesarán los datos y luego se enviará una petición a Elastic MapReduce para iniciar un job Flow.
2	Elastic MapReduce levanta un cluster EC2, que carga y corre Hadoop.
3	Hadoop ejecuta un job flow descargando los datos desde Amazon S3 en el cluster y las instancias esclavos.
4	Hadoop procesa los datos y luego los carga los resultados a Amazon S3.
5	Se recibe una notificación que indica que el job flow ha terminado y que se pueden descargar los datos procesados desde Amazon S3.

Tabla III.- Pasos de Elastic Mapreduce.

2.2.3 Comparación entre las alternativas (EC2 vs. EMR)

A continuación se presenta un análisis de los costos de las alternativas presentadas en las secciones 2.2.1 y 2.2.2. Para ambos casos, los precios fueron calculados considerando el uso de "High CPU Medium instances" de AWS, y S3 para el almacenamiento de los datos.

NOTA: Los costos detallados en esta tabla fueron obtenidos de la Calculadora de Costos AWS[3], y están actualizados al 25 de agosto de 2009.

Número de Gigabytes almacenados en S3:	**15**
Costo por instancia en EC2	0,2
Recargo por uso de Elastic MapReduce, por hora*instancia	0,03
Data transfer in estimado (en GB)	1
Data transfer out estimado (en GB)	1
Duración de una consulta	1

Tabla IV.- Consideraciones tomadas para el análisis.

[3] http://aws.amazon.com

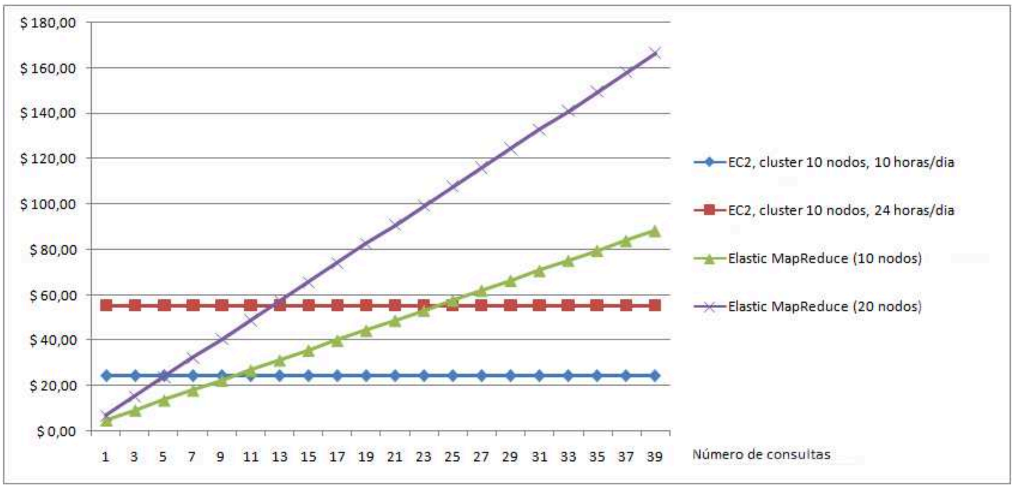

Figura 3. EC2 vs. EMR

La Figura 3 muestra una comparación entre utilizar EC2 con Hadoop vs utilizar EMR. En el gráfico se puede observar que es más conveniente usar EMR si se planea hacer menos de 10 consultas en un lapso de 10 horas, y que si se planea hacer mas de 10 consultas en un lapso de 10 horas es más conveniente usar el esquema de EC2 con 10 nodos /10 horas al día, porque se podría hacer las consultas que se desee y el precio no cambiaría.

2.3 Detalle de las herramientas utilizadas

2.3.1 Hadoop 1.8

Framework de uso libre que soporta aplicaciones distribuidas para grandes cantidades de datos, Esto permite a las aplicaciones trabajar con miles de nodos y Petabytes de datos [8].

2.3.2 AWS (EC2, S3, Elastic Map Reduce)

EC2 Amazon EC2 es sencilla interfaz de servicio web le permite obtener y la capacidad de configurar con mínima fricción. Le proporciona un control completo de sus recursos de computación en la nube[4].

S3 (Simple Storage Service) Es un servicio de almacenamiento masivo, totalmente transparente en el internet, escalable y fácil para los desarrolladores[5].

EMR Amazon Elastic MapReduce (EMR) incrusta automáticamente una implementación del MapReduce framework (Hadoop) en las instancias de Amazon EC2, sub-dividiendo los datos de un flujo de trabajo en pequeñas partes.

2.3.3 Lenguajes de programación y librerías

Para el desarrollo de la interfaz Web de este proyecto se utilizó el lenguaje de programación PHP versión 5.2.6 y AJAX. Para la manipulación de los datasets de la Wikipedia se utilizó la librería Cloud[9].

2.3.4 Diseño de la solución del proyecto

La Figura 4 muestra el diagrama de interacción del usuario final con la interfaz Web que permite levantar de manera asíncrona un cluster EC2 por medio del uso del servicio Elastic MapReduce. Los datasets de la Wikipedia y el jar que contiene el algoritmo MapReduce se encuentran almacenados en Amazon S3, los resultados del `grep` se

[4] http://aws.amazon.com/ec2/

[5] http://aws.amazon.com/s3/

almacenan en un bucket S3, posteriormente la interfaz Web muestra una lista de los resultados con enlaces a la página de la Wikipedia, de cada artículo donde haya encontrado una coincidencia de la expresión regular.

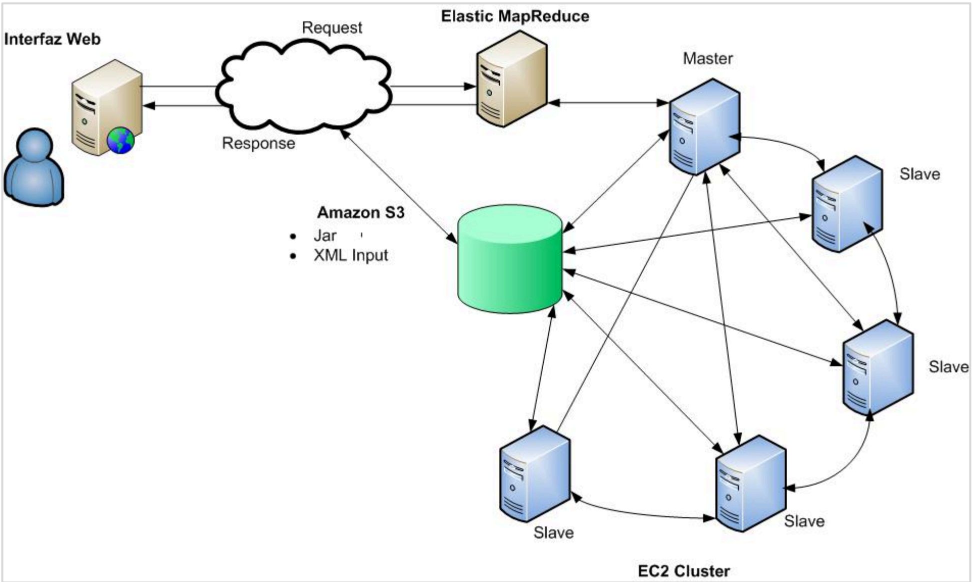

Figura 4. Diagrama de la solución

La Figura 5 muestra la implementación del algoritmo MapReduce para este proyecto. Cada mapper recibe como parámetro de entrada una página de la Wikipedia donde busca las coincidencias de la expresión regular ingresada por el usuario, y luego genera una salida en la cual incluye como clave única la coincidencia más el nombre del artículo y como valor el número de veces que se repite la coincidencia dentro de un artículo. Para el grep se utiliza un reducer que concatena los resultados de los mappers. Luego de esto, la salida del grep se utiliza como entrada para rankear (sort) los resultados.

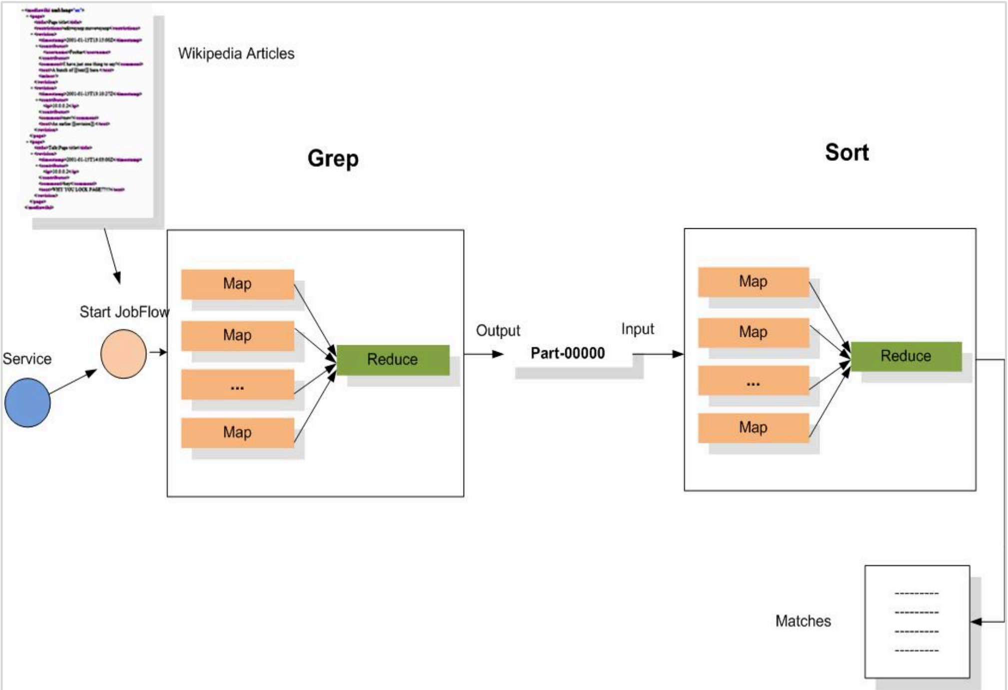

Figura 5. Diagrama algoritmo MapReduce

2.3.5 Pseudocódigo: Mapper, Reducer

Para la presentación del pseudocódigo de la solución, se ha utilizado la nomenclatura estándar para algoritmos MapReduce[9]. El pseudocódigo presentado a continuación ilustra de manera concreta, los pasos descritos en la sección anterior.

```
GrepMapper
	(docid,wikipediaPages)->[(match,[docid,1])]
GrepReducer
	(match,[docid1,docid2,docid1…])->[(match,[docid1,n1]),
	(match,[docid2,n3]),…..]
SortMapper
	(docid,n) -> [(n,docid)]
SortReducer
	(n,docid) ->  [(docid,n)]
```

Figura 6. Pseudocódigo del Algoritmo

2.4 Diseño de la interfaz

A continuación presentamos el diseño de la interfaz del sistema propuesto para interactuar con el usuario. Se define la forma de realizar las consultas y la información mostrada, resultados del sistema.

Figura 7. Interfaz Web

Se ha optado por el diseño minimalista de interfaces de máquinas de búsqueda, popularizado por Google. Este diseño permite que el usuario pueda interactuar con el motor de búsqueda de manera sencilla, ya que utiliza una interfaz a la que él ya está acostumbrado. Los resultados se muestran a manera de enlaces a los artículos de la Wikipedia que concuerdan con la expresión regular, indicando también, el número de veces que la expresión aparece en la página.

Capítulo 3

3. Resultados

3.1 Introducción

En este capítulo detalla los resultados obtenidos en las pruebas realizadas. Se incluye cada uno de los factores que afectan directamente a la ejecución del `grep` y la salida que este genera. Cada resultado fue analizado para su posterior optimización. Para las pruebas se realizó una selección varios patrones de expresiones regulares las cuales al ser consultadas en páginas como Google o la misma Wikipedia no se obtuvo resultado satisfactorio.

Los caracteres disponibles para el uso del Wikigrep distribuido son definidos por la implementación del paquete nativo de java `java.util.regex` [10].

#	Expresión	Descripción	Coincidencias
1	"(\\d\\d\\d\\d)-(\\d\\d\\d\\d)"	Hallar fechas dentro de un rango	Antonio Lucio Vivaldi (1678)-(1741)
2	"[^\\s]*(less|ness|able)"	Hallar palabras con los sufijos less, ness y able.	Sleepless, capable, greatness
3	"(.)(.).\\2\\1"	Hallar palabras palíndromas de 5 letras	Radar, kayak, level

Tabla V Patrones de Búsqueda

3.2 Resultados

Expresión	# nodos EMR	# de Mappers	# de Reducers	GB	Tiempo Ejecución
1	10	16	1	23	10 min
2	10	16	1	23	11 min
3	10	16	1	23	10 min

Tabla VI Cuadro de Resultados

La Tabla VI muestra un resumen de los resultados del `grep` para cada una de las expresiones listadas anteriormente, donde se muestra el número de nodos utilizados, el número de mappers, el número de reducers el total en GB de los datasets y el tiempo tomado para cada consulta.

Expresión	Google	Wikipedia Search	Wikigrep
"(\\d\\d\\d\\d)-(\\d\\d\\d\\d)"	No es posible	No es posible	Si
"[^\\s]*(less|ness|able)"	No es posible	Es posible por el uso de wildcards ejm: *less, *ness, *able lo que representa un total de 3 consultas	Si
"(.)(.).\\2\\1"	No es posible	No es posible	Si

Tabla VII Comparación entre motores de búsqueda

Tabla VII muestra una comparación entre los motores; Google[6] y Wikipedia Search[7] contra Wikigrep, las búsquedas no son posibles

[6] www.google.com

[7] en.wikipedia.org/wiki/index.php?curid=42133

para las expresiones en el caso de Google, y para el caso de Wikipedia Search solo para la expresión **“[^\\s]*(less|ness|able)”,** que puede ser usada como *less, *ness y *able por separado dado que permite el uso del wildcard “*”, para consultar por los tres sufijos son necesarias tres consultas. Wikigrep es capaz de procesar las 3 consultas anteriores en un tiempo de 10 minutos para un dataset de 23 GB. Este tiempo puede ser reducido a menos de 5 minutos si se utiliza un cluster EC2, en lugar de EMR, ya que al usar EMR se debe levantar y bajar el clúster por cada consulta.

Conclusiones y Recomendaciones

De acuerdo con los objetivos de la investigación que se plantearon al inicio de este proyecto se puede concluir y recomendar lo siguiente:

CONCLUSIONES

1) El uso de computación distribuida se vuelve cada vez más popular gracias al desarrollo de servicios de empresas como Amazon, Google y Microsoft. Sin embargo subir los datos a la nube es aún un problema debido a limitantes como el ancho de banda del usuario.
2) Al trabajar con clústers para el procesamiento masivo de datos, se puede llegar a reducir los tiempos de procesamiento de los mismos considerablemente. Para este proyecto se utilizó datasets de la Wikipedia en inglés que en conjunto llegaron a pesar 23 GB. Las consultas llegaron a durar entre 10 a 15 minutos, lo cual es un tiempo reducido comparado con un grep tradicional que hubiera tomado unas cuantas horas de procesamiento.
3) El desarrollo del Wikigrep distribuido fue un éxito y atribuimos este suceso ha distintas razones. Primero, el diseño de la solución usando como base EMR que nos ayuda en el proceso de levantar un clúster EC2, por otra parte la librería cloud9 que facilito el uso y manipulación de los artículos de la Wikipedia. Tercero la optimización realizada mediante la investigación en el uso del número adecuado de mappers y reducers para cada uno de los algoritmos tanto el grep como el sort.

4) El uso de expresiones regulares es muy importante cuando se busca referencias exactas en un documento o buscar que el texto tenga un patrón y su buen uso puede llegar a ser una gran herramienta para el usuario. Servicios de cloud computing facilitan el desarrollo de herramientas de este tipo que disminuyan los tiempos de procesamiento ahorrando así dinero y tiempo del usuario.

RECOMENDACIONES

1) Recomendamos para versiones futuras el uso de datasets de la Wikipedia comprimidos mejorando así los tiempos de acceso a los archivos de entrada y a su vez el tiempo de respuesta.
2) Actualmente EMR no soporta el uso de los dataset públicos de la Wikipedia, por lo que podría usarse el esquema tradicional EC2 y adjuntar un EBS (Elastic Block Store) con este dataset público y así evitar subir los datos a S3.
3) Una mejora para el orden en que se muestran los resultados es utilizar algún algoritmo de Page Rank que permita mostrar al usuario como resultado las páginas que son más citadas en otras páginas o que son más visitadas como primeras opciones.

Referencias Bibliográficas

1. Giles, G. Internet encyclopaedias go head to head. http://www.nature.com/nature/journal/v438/n7070/full/438900a.html, última visita 27/08/09

2. Wikipedia:Database download http://en.wikipedia.org/wiki/index.php?curid=68321, última visita 27/08/09

3. A. Emtage and P. Deutsch, "Archie--An Electronic Directory Service for the Internet." *Proc. Usenix Winter 1992 Tech. Conf.*, Usenix Assoc., Berkeley, Calif., 1992, pp. 93-110.

4. Wikipedia Searching, http://en.wikipedia.org/wiki/Help:Searching, ultimo acceso 27/06/09

5. Wikipedia Datasets for the Hadoop Hack, http://www.cloudera.com/hadoophack/datasets/wikipedia, último acceso 15/06/09

6. Cloud^{9}, A MapReduce Library for Hadoop, http://www.umiacs.umd.edu/~jimmylin/cloud9/docs/ , último acceso 15/06/09

7. Introducción a Amazon MapReduce, http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/index.html?CHAP_Client.html , último acceso 15/07/09

8. Wiki Hadoop, http://wiki.apache.org/hadoop/ProjectDescription, último acceso 13/08/09

9. J. Dean and S. Ghemawat, " MapReduce: Simpli_ed Data Processing on Large Clusters" Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, Diciembre, 2004.
10. Java tutorial, Regex Expressions, Predefined Character Clases, http://java.sun.com/docs/books/tutorial/essential/regex/pre_char_classes.html , último acceso 27/08/09