"WikiGrep Distribuido: Búsquedas avanzadas en la Wikipedia"

Irene Varas, Gabriel Paladines y Cristina Abad
Facultad de Ingeniería Eléctrica y Computación
Escuela Superior Politécnica del Litoral
Campus Gustavo Galindo Velasco, 09-01-5863, Guayaquil, Ecuador
ivaras@fiec.espol.edu.ec, gpaladin@fiec.espol.edu.ec, cabad@fiec.espol.edu.ec

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 (uno de los servicios de los Amazon Web Services), realiza la búsqueda del patrón dentro de todos los documentos, obtiene la respuesta y la 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, de el Hadoop framework y de datasets de la Wikipedia previamente cargados en Amazon.

Se realizaron pruebas de búsquedas con varias expresiones regulare, 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.

Palabras Claves: *Haddop, cloud computing,MapReduce, Elastic MapReduce, Simple storage service S3,Wikipedia, dataset, clúster EC2.*

Abstract

In this project we created a regular expressions search engine that uses the Wikipedia database of articles. The system allows the use of to enter a regular expression and makes an asynchronous request to initialize an EC2 cluster; it searches for the pattern inside all the Wikipedia and then returns the result, displaying a list of all the occurrences of the pattern and a link to the Wikipedia Article.

We used the Amazon Web Services, Java libraries to manipulate Wikipedia Articles, the Hadoop framework and a dataset of the Wikipedia Articles.

We tested some regular expressions that couldn't be searched for using neither traditional search engines nor the Wikipedia Search Engine.

Our tests show that an advanced search engine could be cheap to implement providing high scalability through the use of cloud computing and data-intensive computing techniques.

1. Introducción

La Wikipedia 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. 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)[1]. Debido a la gran cantidad de información que contiene 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.

2. Análisis de la Plataforma de Desarrollo

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)[3] para levantar clústeres computacionales bajo demanda. Para este proyecto se consideró las dos alternativas descritas a continuación.

2.1 Amazon EC2 , Hadoop y Amazon S3

Levantar un clúster de computadoras utilizando el servicio EC2 (Elastic Computing Cloud)[4] con Hadoop instalado, de n nodos y utilizar el servicio S3 (Simple Storage Service)[5] 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 Elastic Map Reduce

Amazon Elastic MapReduce (EMR)[6] incrusta automáticamente una implementación del MapReduce framework (Hadoop)[7] 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[8].

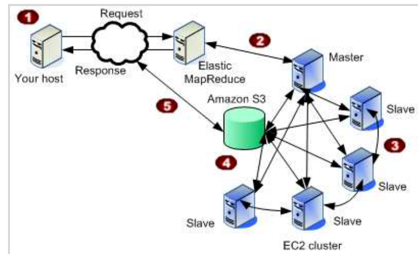

Figura 1. Arquitectura Elastic MapReduce.

Tabla 1. Pasos de Elastic MapReduce.

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.

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 0 y 0. 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.

Tabla 2. Consideraciones tomadas para el Análisis.

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

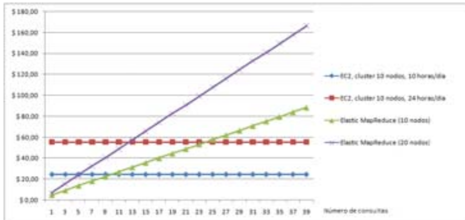

Figura 2. EC2 vs EMR

La Figura 2 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.

3. Formato de los datos

La entidad página contiene 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]].

3.1 Un ejemplo del archivo XML

Figura 3. Formato XML artículo de la Wikipedia.

3.2. 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" [9]; en el que describen 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 es 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 Cloud[9], que describiremos con más detalle a continuación.

4. Diseño de la solución del proyecto

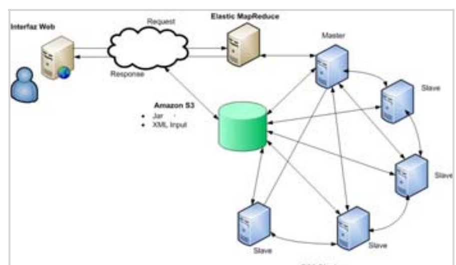

Figura 4. Diagrama de la solución.

La **¡Error! No se encuentra el origen de la referencia.** 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 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.

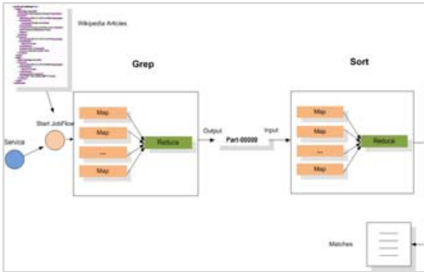

Figura 5. Diagrama algoritmo MapReduce.

La **¡Error! No se encuentra el origen de la referencia.** 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.

4.1. Pseudocódigo: Mapper, Reducer

Para la presentación del pseudocódigo de la solución, se ha utilizado la nomenclatura estándar para algoritmos MapReduce[10]. 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)]
```

4.2 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 6. 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.

5. Resultados

Tabla 3. Patrones de Búsqueda

#	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|nes s|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 4. Cuadro de Resultados

# Ex	# 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

La Tabla 4 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.

Tabla 5. Comparación entre motores de Búsqueda.

# Ex	Google	Wikipedia Search	Wikigrep
1	No es posible	No es posible	Si
2	No es posible	Es posible por el uso de wildcards ejm: *less, *ness, *able lo que representa un total de 3 consultas	Si
3	No es posible	No es posible	Si

La **¡Error! No se encuentra el origen de la referencia.**muestra una comparación entre los motores; Google y Wikipedia Search contra Wikigrep, las búsquedas no son posibles 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.

6. Conclusiones

- 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.
- Al trabajar con clústeres 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.
- 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.
- 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.

6. Recomendaciones

- 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.
- 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.
- 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.

12. Referencias

[1] Giles, G. Internet encyclopaedias go head to head. Fecha de última visita 27/08/09. Disponible en http://www.nature.com/nature/journal/v438/n7070/full/438900a.html
[2] Wikipedia:Database download. Disponible en http://en.wikipedia.org/wiki/index.php?curid=68321. Fecha de última visita 27/08/09.
[3] Amazon Web Services, http://aws.amazon.com. Fecha de última visita 15/09/09.

[4] Amazon EC2, http://aws.amazon.com/ec2/. Fecha de última visita 15/09/09.
[5] Amazon S3, http://aws.amazon.com/s3/. Fecha de última visita 15/09/09.
[6] Amazon EMR, http://aws.amazon.com/elasticmapreduce/, Fecha de última visita 15/09/09
[7] Hadoop Framework, http://hadoop.apache.org/. Fecha de última visita 15/09/09.
[8] Introducción a Amazon MapReduce. http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/index.html?CHAP_Client.html. Fecha de última visita 15/07/09.
[9] Wikipedia Datasets for the Hadoop Hack, http://www.cloudera.com/hadoophack/datasets/wikipedia. Fecha de última visita 15/06/09.
[10] 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.