Tecnológico de Costa Rica

TEC

Mini Buscador Web

Manual de Usuario

Bryan Steve Jiménes Chacón – 2014114175
Roberto Fernández Valerio - 2013010399

Tabla de Contenidos

Introducción

El objetivo crear un buscador web es ayudar a entender como es una estrategia para afrontar el reto que presenta una aplicación de uso diario en la web como lo es un buscador en el internet de hoy en día, que procesa tanta información que no permite una búsqueda directa, sino que nos obliga a buscar otros métodos para manejar tanta cantidad de datos, esto lo podemos lograr con el Map-Reduce de Hadoop, una tecnología de Google que nos ayuda a procesar enormes cantidades de datos y en nuestro caso específico a identificar la ubicación de los textos en la web para poder indexarlos.

Manual de Usuario

¡Gracias por utilizar nuestro Mini Buscador Web!

¡Atención! Este manual debe ser leído atentamente antes de proceder a usar este software. El no seguir estas instrucciones puede dar como resultado información no veraz. Dedíquele unos minutos en familiarizarse con su nuevo software.

Este manual fue creado con el propósito de que el usuario final llegue a saber cómo utilizar nuestro Mini Buscador Web en la forma más sencilla posible.

Herramientas necesarias para el uso del programa

Para hacer posible el uso de este programa son necesarios:

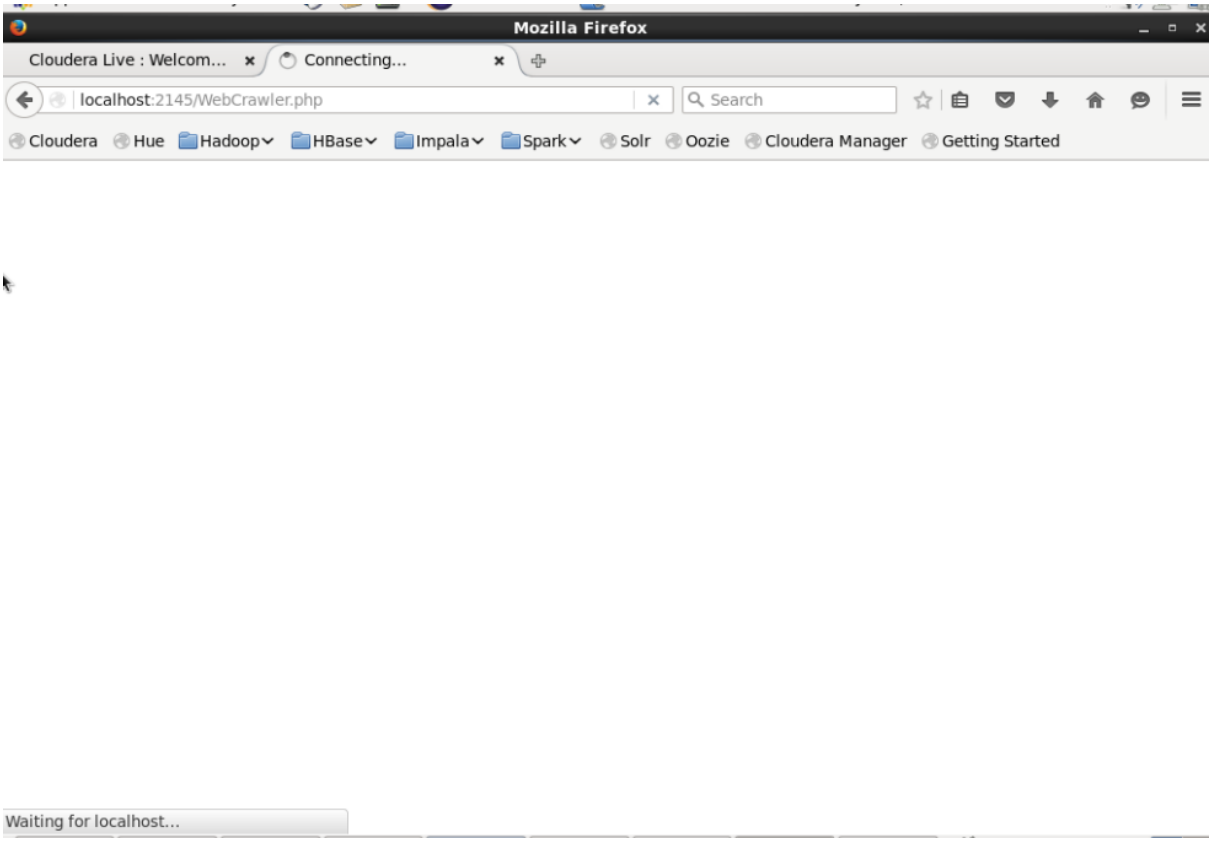
- Una computadora con los siguientes requerimientos mínimos:
  - Máquina Virtual de Cloudera
  - 8GB de RAM
  - Disco duro de 100GB
  - Procesador Intel® i5 o i7 de cuarta generación o equivalente
  - Tarjeta de Video de 512MB o superior
  - Conexión a internet de banda ancha
- Es necesario un navegador para poder acceder al sitio web
- Máquina virtual Cloudera QuickStart, ya que tiene Hadoop y MySQL incorporado

Compilación y Ejecución

A continuación, presentaremos una forma práctica de observar el funcionamiento map-reduce a partir de un ejemplo con un buscador web, el cual se divide en tres secciones importantes, un crawler que recoge los datos de una gran cantidad de sitios web, el uso de hadoop para analizar los datos y convertirlos en algo que se pueda almacenar y consultar en el buscador por medio de php.

Primero que todo se debe instalar hadoop y una base de datos por ejemplo MySQL en nuestra máquina, debido a la complejidad de instalar y configurar todo lo necesario, podemos descargar e instalar la máquina virtual de cloudera en http://www.cloudera.com/downloads.html , esta máquina virtual ya tiene casi todo lo que necesitaremos funcionando, es importante recalcar que para acceder a la base de datos se utiliza el usuario “root” y la clave “cloudera”. Además necesitamos instalar algún servidor web por ejemplo podemos utilizar XAMPP para de esta forma poder ejecutar nuestro buscador localmente con estas preparaciones básicas comenzaremos a ejecutar el programa.

Primero es importante tener el archivo WebCrawler.php en la carpeta htdocs, en el caso de XAMPP, o la correspondiente para el servidor web, y nos disponemos a acceder a la misma desde nuestro navegador con la url localhost:”puerto”/WebCrawler.php



Este proceso creará un archivo para que posteriormente hadoop lo lea, debido a que el objetivo es trabajar con grandes cantidades de datos la creación de archivos puede tardar varios minutos.

```
/·/Policies and Legal Information - W3C
/-/http://www.w3.org/
/+/ &#xAE; W3C is pleased to announce the selection of Web Scienc
/*/
/·/World Wide Web Consortium (W3C)
/-/http://www.csail.mit.edu/
/+/ Cybersecurity @ MIT CSAIL bigdata@CSAIL Robotics Wireless@MIT
/*/
/·/MIT Computer Science and Artificial Intelligence Laboratory | I
/-/http://www.ercim.org/
/+/ Interested in joining ERCIM? - download the flyer in pdf  H20
/*/
/·/ERCIM - the European Research Consortium for Informatics and M
/-/http://www.keio.ac.jp/
/+/ "To understand the field, and learn from the field" A new web
/*/
/·/Keio University
/-/http://www.w3.org/Consortium/Legal/ipr-notice
/+/ W3C's Intellectual Rights Notices and Legal Disclaimers &#xAE
/*/
/·/Policies and Legal Information - W3C
/-/http://www.w3.org/Consortium/Legal/ipr-notice
/+/ W3C's Intellectual Rights Notices and Legal Disclaimers &#xAE
/*/
/·/Policies and Legal Information - W3C
/-/http://www.w3.org/Consortium/Legal/copyright-documents
/+/ ALL NOTICE If it exists Status: &#xAE; Status: This document
```

El resultado tendrá miles de líneas por lo que no sería posible para un humano analizar tantas palabras y un programa normal tardaría demasiado, es aquí donde entramos a la parte principal del proyecto, donde se aprecian las propiedades y el uso de hadoop.

En primer lugar, recordemos que ya tenemos hadoop funcionando, por lo que podemos pasar directamente a crear los directorios para los filesystems que vamos a procesar y generar, para esto ejecutamos:

sudo su hdfs
hadoop fs -mkdir /user/cloudera

hadoop fs -chown cloudera /user/cloudera
Exit

sudo su cloudera
hadoop fs -mkdir /user/cloudera/job1 /user/cloudera/job1/input

sudo su cloudera
hadoop fs -mkdir /user/cloudera/job2 /user/cloudera/job2/input

sudo su cloudera
hadoop fs -mkdir /user/cloudera/job3 /user/cloudera/job3/input

Ahora que tenemos el destino para el resultado de nuestro WebCrawler nos vamos a la carpeta donde se encuentra el crawl y ejecutamos lo siguiente:

hadoop fs -put crawl /user/cloudera/job1/input

hadoop fs -put crawl /user/cloudera/job2/input

hadoop fs -put crawl /user/cloudera/job3/input

Ahora nos queda ejecutar los jobs, para esto nos debemos ir a la ubicación de cada job y ejecutar los siguiente comandos:

Job1:

hadoop fs -rm -r /user/cloudera/job1/output

mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* Job1.java -d build -Xlint

jar -cvf Job1.jar -C build/ .

hadoop jar Job1.jar org.myorg.Job1 /user/cloudera/job1/input /user/cloudera/job1/output

Job2:

hadoop fs -rm -r /user/cloudera/job2/output

mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* Job2.java -d build -Xlint

jar -cvf Job2.jar -C build/ .

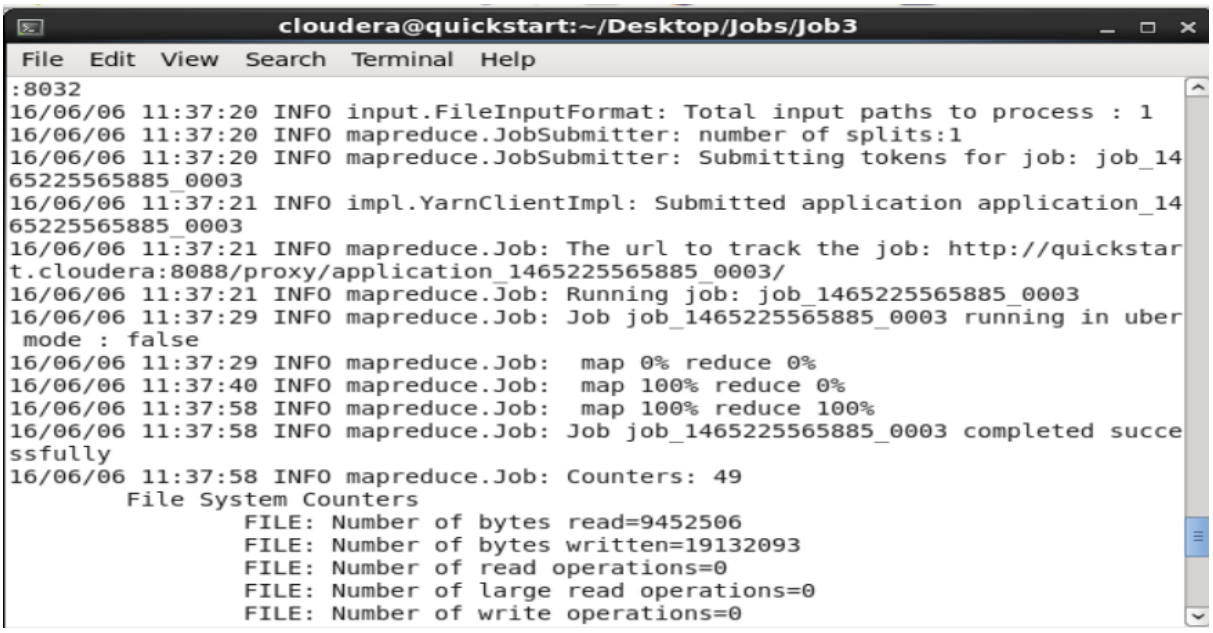
hadoop jar Job2.jar org.myorg.Job2 /user/cloudera/job2/input /user/cloudera/job2/output

Job3:

hadoop fs -rm -r /user/cloudera/job3/output
mkdir -p build
javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* Job3.java -d build -Xlint
jar -cvf Job3.jar -C build/ .
hadoop jar Job3.jar org.myorg.Job3 /user/cloudera/job3/input /user/cloudera/job3/output

Esto nos mostrará un mensaje al final con el resumen del resultado del procesamiento de los jobs. Ahora ya tenemos un output con nuestro map-reduce ejecutado, lo que sigue

es ejecutar un programa que se encargue de leer el resultado e insertarlo en la base de datos, pero primero debemos asegurarnos de que nuestra base de datos MySQL tenga las tablas creadas.



En este caso creamos una base de datos llama proyecto y dentro ejecutamos los siguientes comandos para crear las tablas:

mysql> create table job1 (Palabra varchar(255), Url varchar(255));

mysql> create table job2 (Palabra varchar(255), Url MediumText,Conteo int);

mysql> create table job3 (Palabra varchar(255), Conteo int);

Exit

Con nuestras tablas creadas simplemente procederemos a ir a la ubicación de la carpeta Procesamiento para cada Job y ejecutamos los siguientes comandos

Job1:
mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* DatosJob1.java -d build -Xlint

jar -cvf DatosJob1.jar -C build/ .

hadoop jar DatosJob1.jar DatosJob1

Job2:
mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* DatosJob2.java -d build -Xlint

jar -cvf DatosJob2.jar -C build/ .

hadoop jar DatosJob2.jar DatosJob2

Job3:

mkdir -p build

javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* DatosJob3.java -d build -Xlint

jar -cvf DatosJob3.jar -C build/ .

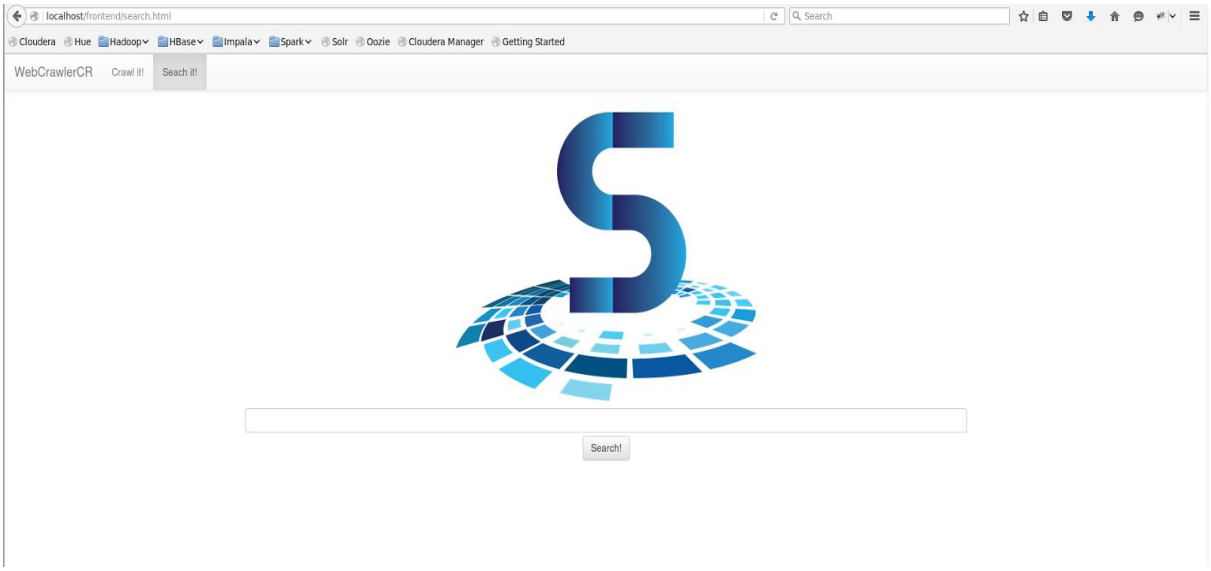
hadoop jar DatosJob3.jar DatosJob3

Estos comandos ejecutarán un programa que se conecta a la base de datos proyecto con el usuario root y la clave cloudera e insertará en las tablas correspondientes a los jobs los resultados del output de hadoop ya procesados para poder ser consultados en un futuro por el buscador.

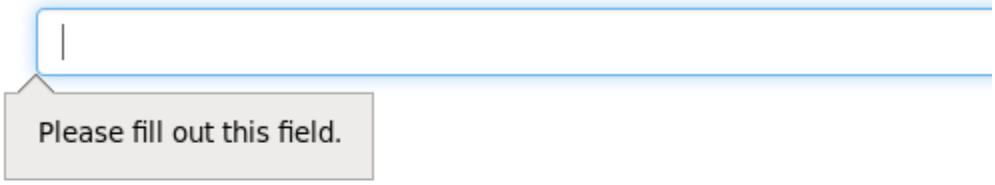
Buscar

Ahora ya podemos realizar consultas desde el buscador, basta con ir a la dirección donde tengamos la página dentro de htdocs desde el navegador y realizar las búsquedas.

Dentro de la misma página donde se ubica el Web Crawler, se puede ver la pestaña de navegación de Búsqueda, en el cual tiene la siguiente interfaz:



El mismo se pueden buscar una o varias palabras, siempre y cuando las mismas sean mayores a dos caracteres. Asimismo, este no puede tener un campo vacío.

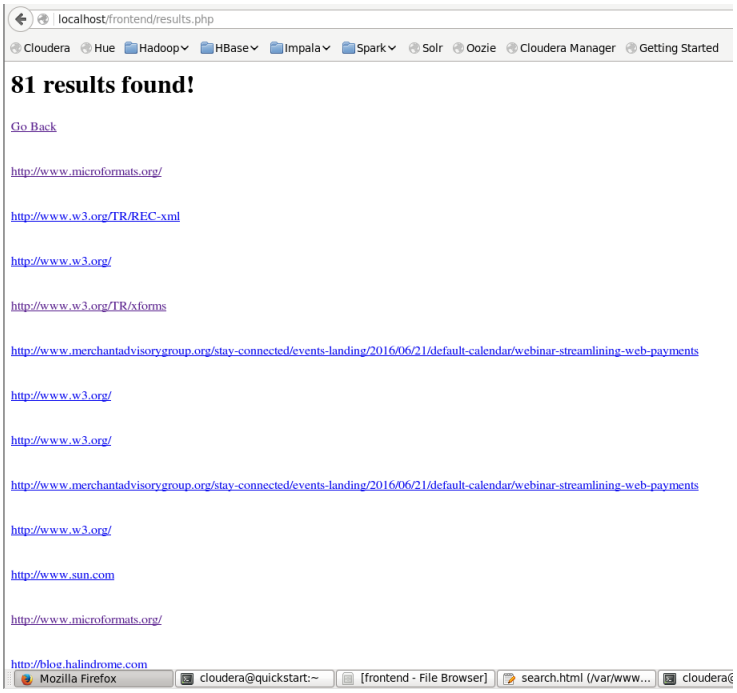


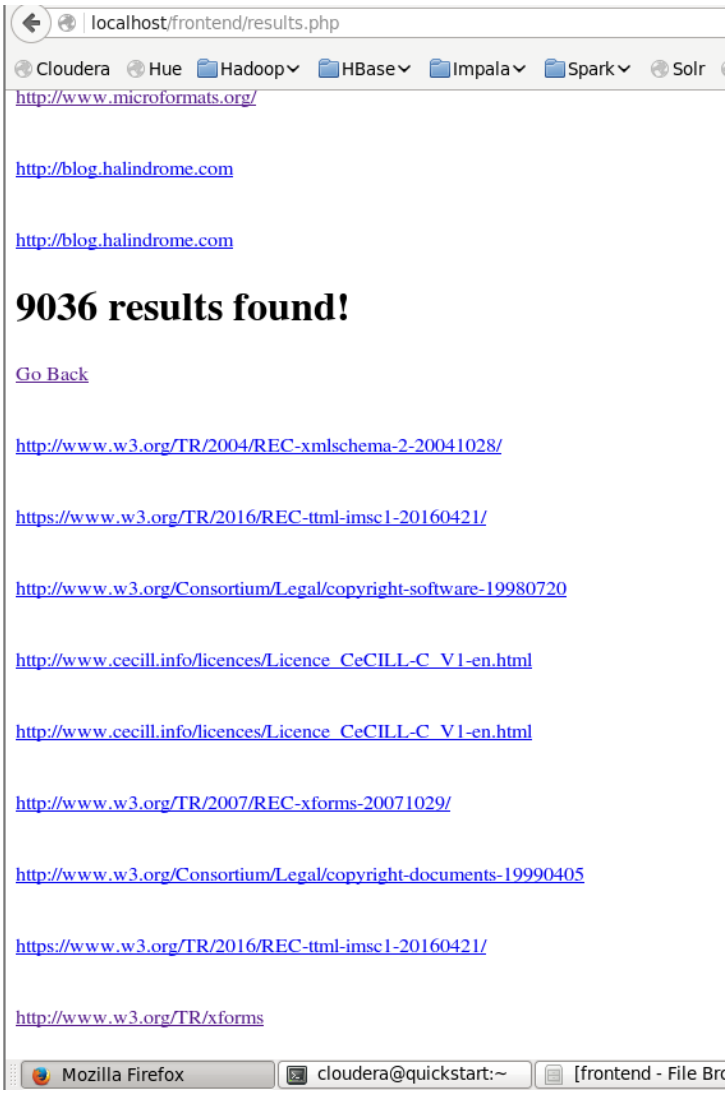
Cuando se ejecuta la búsqueda, se buscarán las palabras de manera individual, y este mostrará los links de las páginas que contienen las mismas.

Tecnológico de Costa Rica, Centro Académico de San José

Bases de Datos II

6 de junio del 2016




localhost/frontend/results.php
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
http://www.microformats.org/
http://blog.halindrome.com
http://blog.halindrome.com
9036 results found!
Go Back
http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/
https://www.w3.org/TR/2016/REC-ttml-imsc1-20160421/
http://www.w3.org/Consortium/Legal/copyright-software-19980720
http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html
http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html
http://www.w3.org/TR/2007/REC-xforms-20071029/
http://www.w3.org/Consortium/Legal/copyright-documents-19990405
https://www.w3.org/TR/2016/REC-ttml-imsc1-20160421/
http://www.w3.org/TR/xforms
Mozilla Firefox
cloudera@quickstart:~
[frontend - File Br

Conclusión

Con este ejemplo pudimos ver de manera básica que es lo que se necesita para ejecutar un map-reduce y poder implementar la solución en un proyecto con aplicación real, este mini-buscador web puede alcanzar mucha más potencia y características fácilmente si seguimos implementando más funciones con hadoop para procesar datos más finamente y mejores resultados para las búsquedas por lo que hadoop se muestra como una opción que ofrece grandes resultados con poco código y nos ahorra mucho tiempo así q aumenta la eficiencia para todos.