Big data de código abierto para el impaciente, Parte 1: Tutorial Hadoop: Hello World con Java, Pig, Hive, Flume, Fuse, Oozie, y Sqoop con Informix, DB2, y MySQL

Cómo iniciar con Hadoop y sus bases de datos favoritas

Marty Lurie (lurie@us.ibm.com)
Systems Engineer
IBM Corporation

20-05-2013

Este artículo está enfocado en explicar Big Data y posteriormente proporcionar ejemplos sencillos trabajados en Hadoop, el más importante jugador de código abierto en el espacio de Big Data. A usted le agradará escuchar que Hadoop NO es un reemplazo para Informix® o DB2®sino que interactúa muy bien con la infraestructura existente. Existen múltiples componentes en la familia Hadoop y este artículo detallará muestras de código específico que muestran las posibilidades. No habrá estampida de elefantes si usted prueba estos ejemplos en su propia PC.

Hay mucha emoción con relación a Big Data, pero también mucha confusión. Este artículo proporcionará una definición de trabajo de Big Data y posteriormente pasará por algunas series de ejemplos para que usted puede tener una comprensión de primera mano de algunas de las posibilidades de Hadoop, la tecnología líder en código abierto dentro del dominio de Big Data. Enfoquémonos específicamente en las siguientes preguntas.

- ¿Qué es Big Data, Hadoop, Sqoop, Hive, y Pig, y por qué hay tanta emoción en relación a esta área?
- ¿Cómo se relaciona Hadoop a IBM DB2 y a Informix? ¿Estas tecnologías pueden funcionar juntas?
- ¿Cómo puedo iniciar con Big Data? ¿Cuáles son algunos ejemplos sencillos que se ejecutan en una única PC?
- Para el super impaciente, si ya puede definir Hadoop y desea iniciar inmediatamente con ejemplos de código, entonces haga lo siguiente.
    1. Ponga en marcha su instancia Informix o DB2.

2. Descargue la imagen VMWare del Sitio Web Cloudera e incremente el valor de RAM de máquina virtual a 1,5 GB.
3. Salte hasta la sección que contiene ejemplos de código.
4. Existe una instancia MySQL incorporada en la imagen VMWare. Si usted está realizando ejercicios sin conectividad de red, use ejemplos MySQL.

Para todos los demás, continúen leyendo...

¿Qué es Big Data?

Big Data son grandes en cantidad, se capturan a un índice rápido, y son estructurados o no estructurados, o alguna combinación de lo de arriba. Estos factores hacen que los Big Data sean difíciles de capturar, extraer, y gestionar usando métodos tradicionales. Hay tanta publicidad en esta espacio que podría haber un amplio debate tan solo en relación a la definición de big data.

El uso de la tecnología Big Data no se restringe a grandes volúmenes. Los ejemplos en este artículo usan pequeños ejemplos para ilustrar las posibilidades de la tecnología. A partir del años 2012, los clústers que son *grandes* están en el rango de 100 Petabyte.

Los Big Data pueden ser tanto estructurados como no estructurados. Las bases de datos relacionales tradicionales, como Informix y DB2, proporcionan soluciones comprobadas para datos estructurados. A través de la extensibilidad, también gestionan datos no estructurados. La tecnología Hadoop trae técnicas de programación nuevas y más accesibles para trabajar en almacenamientos de datos masivos con datos tanto estructurados como no estructurados.

¿Por qué toda la emoción?

Existen muchos factores que contribuyen a la publicidad que está alrededor de Big Data, incluyendo lo siguiente.

- Reuniendo el cómputo y el almacenamiento en un hardware de producto: el resultado consiste en una increíble velocidad a un bajo costo.
- Rentabilidad: La tecnología Hadoop proporciona ahorros significativos (piense en un factor de aproximadamente 10) con mejoras en rendimiento significativas (de nuevo, piense en un factor de 10). Su kilometraje puede variar. Si la tecnología existente puede ser tan dramáticamente derrotada, vale la pena examinar si Hadoop puede complementar o reemplazar aspectos de su arquitectura actual.
- Escalamiento Lineal: Cada tecnología paralela asevera el escalamiento hacia arriba. Hadoop tiene escalamiento genuino ya que el último release está *expandiendo el límite del número de nodos a más allá de 4.000*.
- Acceso a datos no estructurados: Un almacén de datos altamente escalable con un buen modelo de programación paralelo, MapReduce, ha representado un desafío para la industria por algún tiempo. El modelo de programación de Hadoop no resuelve todos los problemas, pero es una solución robusta para muchas tareas.

Distribuciones Hadoop: IBM y Cloudera

Uno de los puntos de confusión es, "Dónde obtengo el software para trabajar en Big Data?" Los ejemplos en este artículo se basan en la distribución libre de Cloudera del Hadoop llamado CDH (para distribución de Cloudera incluyendo Hadoop). Esto está disponible como

una imagen VMWare del sitio web de Cloudera. IBM ha anunciado recientemente que está convirtiendo su plataforma de big data para que se ejecute en CDH. Se pueden encontrar más detalles en Recursos .

El término *tecnología disrruptiva* es usado en exceso, pero en este caso puede ser apropiado.

¿Qué es Hadoop?

A continuación hay varias definiciones de Hadoop, cada una dirigida a una audiencia dentro de la empresa:

- Para los ejecutivos: Hadoop es un proyecto de software de código abierto de Apache para obtener valor de volumen/velocidad/variedad increíbles de datos acerca de su organización. Use los datos en vez de desechar la mayoría de ellos.
- Para los gerentes técnicos: Una suite de código abierto de software que extrae los BigData estructurados y no estructurados acerca de su compañía. Se integra con su ecosistema existente de Inteligencia de Negocios.
- Para el departamento legal: Una suite de código abierto de software que es empacado y cuenta con soporte de múltiples proveedores. Vea la sección Recursos en relación a indemnización IP.
- Ingeniería: Un entorno de ejecución paralelo masivamente, de nada compartido, basado en Java map-reduce. Piense en cientos a miles de computadoras trabajando en el mismo problema, con resiliencia a fallas incorporada. Los proyectos en el ecosistema Hadoop proporcionan cargado de datos, lenguajes de alto nivel, despliegue automatizado de nube, y otras posibilidades.
- Seguridad: Una suite de software con seguridad Kerberos.

¿Cuáles son los componentes de Hadoop?

El proyecto Apache Hadoop tiene dos componentes centrales, el almacenamiento de archivos llamado Hadoop Distributed File System (HDFS), y la infraestructura de programación llamada MapReduce. Existen diversos proyectos de soporte que aprovechan HDFS y MapReduce. Este artículo proporcionará un resumen, y lo alienta a obtener el libro de OReily "Hadoop The Definitive Guide", 3a Edición, para obtener más detalles.

Las definiciones de abajo tiene la intención de proporcionar los antecedentes necesarios para usar los ejemplos de código que siguen. Este artículo tiene realmente la intención de iniciarlo con una experiencia práctica con la tecnología. Este es más un artículo de "cómo hacer" que de "qué es" o "discutamos".

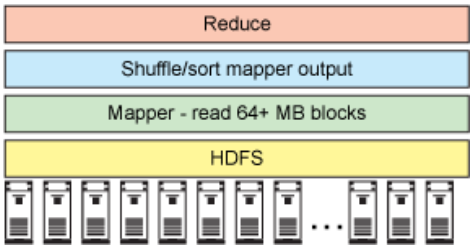
- **HDFS**: Si usted desea más de 4000 computadoras trabajando en sus datos, entonces más vale que distribuya sus datos a lo largo de más de 4000 computadoras. HDFS hace esto para usted. HDFS tiene pocas partes movibles. Datanodes almacena sus datos, y Namenode da seguimiento al lugar donde se almacenan las cosas. Hay otras cuestiones, pero usted ya tiene lo suficiente para iniciar.
- **MapReduce**: Este es el modelo de programación para Hadoop. Existen dos fases, no es de sorprender que se llamen Map y Reduce. Para impresionar a sus amigos dígales que hay un tipo de mezcla entre la fase Map y la fase Reduce. JobTracker gestiona los más de 4000

componentes de su trabajo MapReduce. TaskTrackers toma ordenes de JobTracker. Si le gusta Java entonces codifíquelo en Java. Si a usted le gusta SQL u otro lenguaje que no sea Java tiene suerte, usted puede usar una utilidad llamada Hadoop Streaming.

- **Hadoop Streaming**: Una utilidad para permitir a MapReduce codificar en cualquier lenguaje: C, Perl, Python, C++, Bash, etc. Los ejemplos incluyen un correlacionador Python y un reductor AWK.
- **Hive and Hue**: Si a usted le gusta SQL, estará encantado de escuchar que usted puede escribir SQL y hacer que Hive lo convierta a un trabajo de MapReduce. No, usted no obtiene un entorno ANSI-SQL completo, pero usted obtiene 4000 notas y escalabilidad multi-Petabyte. Hue le brinda una intefaz gráfica basada en navegador para realizar su trabajo Hive.
- **Pig**: Un entorno de programación de nivel alto para realizar codificación MapReduce. El lenguaje Pig es llamado Pig Latin. A usted puede parecerle el nombre poco convencional, pero obtiene rentabilidad y alta disponibilidad increíbles.
- **Sqoop**: Proporciona transferencia de datos bidireccional entre Hadoop y su base de datos relacional favorita.
- **Oozie**: Gestiona flujo de trabajo Hadoop. Esto no reemplaza a su planificador o herramienta BPM, pero proporciona ramificación de "if-then-else" y control dentro de sus trabajos Hadoop.
- **HBase**: Un almacenamiento de valor de clave súper escalable. Funciona similarmente a un hash-map persistente (para los aficionados de python piensen en diccionario). No es una base de datos relacional pese al nombre HBase.
- **FlumeNG**: Un cargador en tiempo real para transmitir sus datos hacia Hadoop. Almacena datos en HDFS y HBase. Usted deseará iniciar con FlumeNG, que mejora el canal original.
- **Whirr**: Suministro de nube para Hadoop. Usted puede arrancar un clúster en unos cuantos minutos con un archivo de configuración muy corto.
- **Mahout**: Aprendizaje de máquina para Hadoop. Usado para análisis predictivos y otros análisis avanzados.
- **Fuse**: Hace que el sistema HDFS parezca como un sistema de archivos normal para que usted pueda usar ls, rm, cd, y otros en datos HDFS
- **Zookeeper**: Usado para gestionar sincronización para el clúster. Usted no estará trabajando mucho con Zookeeper, pero trabaja mucho por usted. Si usted piensa que necesita escribir un programa que use Zookeeper usted es ya sea muy, muy inteligente y podría formar un comité para un proyecto Apache, o usted está apunto de tener un día terrible.

La Figura 1 muestra las piezas claves de Hadoop.

Figura 1. Arquitectura Hadoop



HDFS, la capa inferior, yace sobre un clúster de hardware de producto. Servidores sencillos montados en bastidor, cada uno con 2 núcleos Hex, 6 a 12 discos, y ram de 32 gig. Para un

trabajo de map-reduce, la capa del correlacionador lee a partir de los discos a muy alta velocidad. El correlacionador emite pares de valores claves que son ordenados y presentados al reductor, y la capa de reductor resume los pares de valor clave. No, usted no necesita resumir, usted de hecho puede tener un trabajo map-reduce que solo tiene correlacionadores. Esto debe ser más fácil de comprender cuando usted llegue al ejemplo python-awk.

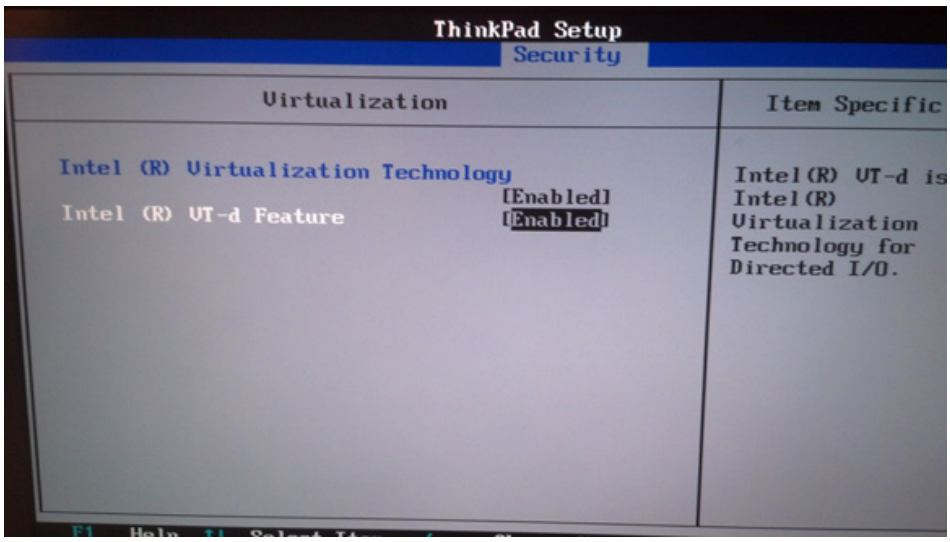
¿Cómo se integra Hadoop con mi infraestructura de Informix o DB2?

Hadoop se integra muy bien con las bases de datos Informix y DB2 con Sqoop. Sqoop es la implementación líder de código abierto para mover datos entre Hadoop y bases de datos relacionales. Usa JDBC para leer y escribir Informix, DB2, MySQL, Oracle, y otras fuentes. Existen adaptadores optimizados para varias bases de datos, incluyendo Netezza y DB2. Vea la sección Recursos para saber cómo descargar estos adaptadores. Los ejemplos son todos específicos para Sqoop.

Iniciando: Cómo ejecutar ejemplos sencillos de Hadoop, Hive, Pig, Oozie, y Sqoop

Ya acabaron las introducciones y definiciones, ahora es momento de lo bueno. Para continuar, ¡usted necesitará descargar VMWare, virtual box, u otra imagen del Sitio Web Cloudera y comenzar a realizar MapReduce! La imagen virtual asume que usted cuenta con una computadora de 64bit y uno de los entornos de virtualización populares. La mayoría de los entornos de virtualización tienen una descarga gratuita. Cuando usted intente arrancar una imagen virtual de 64bit usted puede recibir quejas acerca de las configuraciones BIOS. La Figura 2 muestra el cambio requerido en BIOS, en este caso en una Thinkpad™. Tenga precaución cuando realice los cambios. Algunos paquetes corporativos de seguridad requerirán una contraseña después de un cambio e BIOS antes de que el sistema vuelva a cargar.

Figura 2. Configuraciones BIOS para un huésped virtual de 64bit



Los big data usados aquí son de hecho pequeños. El punto no es hacer que su laptop se incendie por el procesamiento de un archivo masivo, sino mostrar sus fuentes de datos que sean interesantes, y los trabajos map-reduce que responden preguntas significativas.

Descargue la imagen Hadoop virtual

Es muy recomendado que use la imagen Cloudera para ejecutar estos ejemplos. Hadoop es una tecnología que resuelve problemas. El empaque de la imagen Cloudera le permite enfocarse en las preguntas de big-data. Pero si decide ensamblar todas las partes por sí mismo, Hadoop se ha convertido en el problema, no en la solución.

Descargue una imagen. La imagen CDH4, que es la última oferta está disponible aquí: CDH4 image. La versión Previa, CDH3, está disponible aquí: CDH3 image.

Usted cuanta con opciones de tecnologías de virtualización. Usted puede descargar un entorno de virtualización gratuito desde VMWare y otros. Por ejemplo, vaya a vmware.com y descargue el vmware-player. Su laptop probablemente se está ejecutando en Windows así que usted probablemente descargará el vmware-player for Windows. Los ejemplos de este artículo usarán VMWare, y ejecutarán Ubuntu Linux usando "tar" en vez de "winzip" o equivalente.

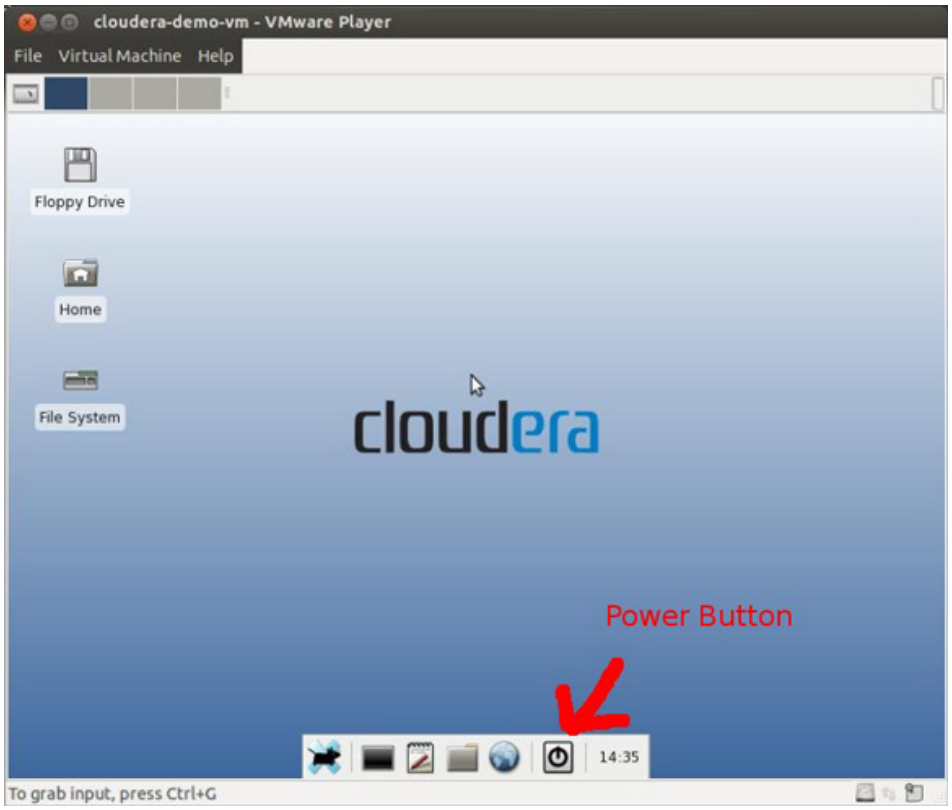
Una vez descargado, realice el proceso untar/unzip como sigue: `tar -zxvf cloudera-demo-vm-cdh4.0.0-vmware.tar.gz`.

O, si usa CDH3, entonces use lo siguiente: `tar -zxvf cloudera-demo-vm-cdh3u4-vmware.tar.gz`

Unzip típicamente funciona en archivos tar. Una vez descomprimido, usted puede arrancar la imagen de la manera siguiente:
`vmplayer cloudera-demo-vm.vmx`.

Ahora usted tendrá una pantalla que se parece a la que se muestra en la Figura 3.

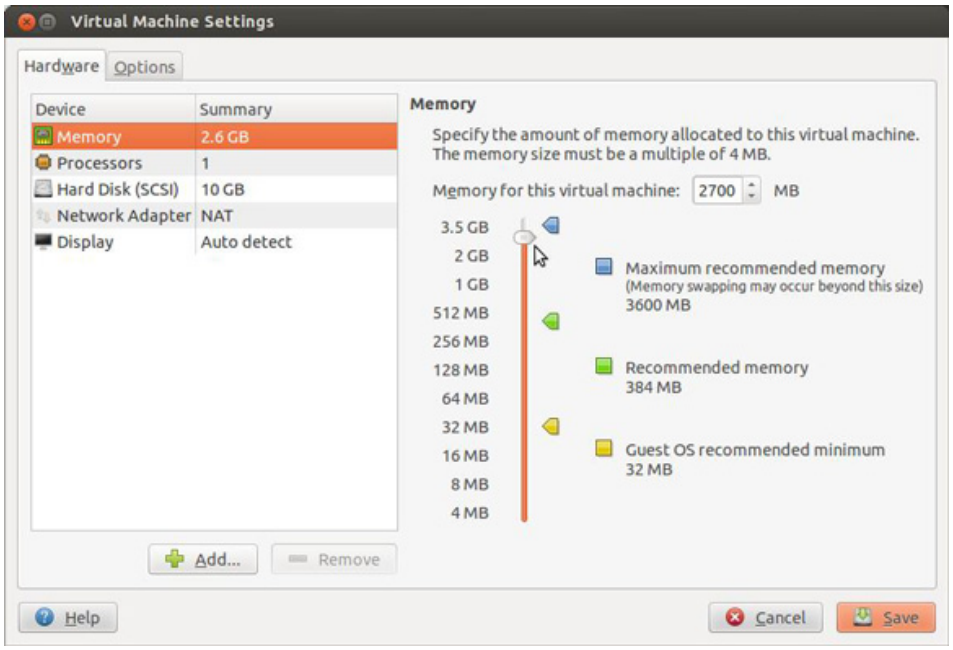
Figura 3. Imagen virtual Cloudera



El comando vmplayer entra de lleno e inicia la máquina virtual. Si usted está usando CDH3, entonces necesitará apagar la máquina y cambiar las configuraciones de memoria. Use el ícono del botón de alimentación que está cerca del reloj en la parte media inferior de la pantalla para apagar la máquina virtual. Posteriormente tiene acceso a editar las configuraciones de la máquina virtual.

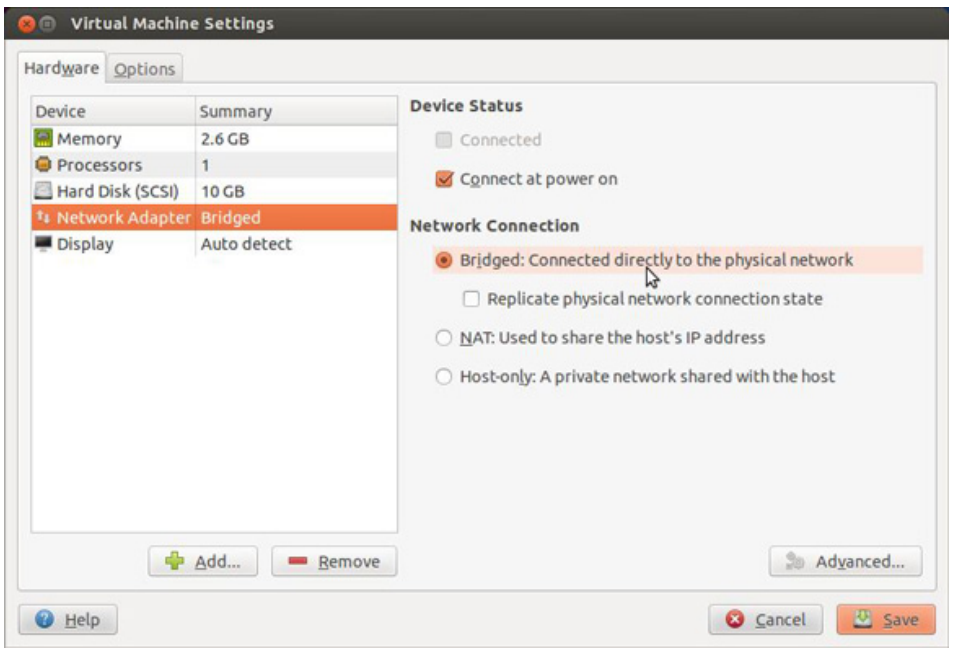
Para CDH3 el próximo paso es super-cargar la imagen virtual con más RAM. La mayoría de las configuraciones solo pueden ser cambiadas con la máquina virtual apagada. La Figura 4 muestra cómo acceder a las configuraciones e incrementar el RAM asignado a más de 2GB.

Figura 4. Añadiendo RAM a la máquina virtual



Como se muestra en la Figura 5, usted puede cambiar las configuraciones de la red para **bridged**. Con esta configuración la máquina virtual obtendrá su propia dirección IP. Si esto crea problemas en su red, entonces usted puede opcionalmente usar Network Address Translation (NAT). Usted estará usando la red para conectarse a la base de datos.

Figura 5. Cambiando las configuraciones de la red a bridged



Usted está limitado por el RAM en el sistema host, así que no intente asignar más RAM de lo que hay en su máquina. Si lo hace, la computadora trabajará muy lentamente.

Ahora, el momento por el que ha estado esperando, encienda la máquina virtual. El usuario cloudera inicia automáticamente sesión en el inicio. Si la necesita, la contraseña Cloudera es: **cloudera**.

Instalación de Informix y DB2

Usted necesitará una base de datos con la cual trabajar. Si usted todavía no tiene una base de datos, puede descargar Informix Developer edition aquí, o gratuitamente DB2 Express-C Edition.

Otra alternativa para instalar DB2 es descargar la imagen VMWare que ya tiene instalado DB2 en un sistema operativo SuSE Linux. Inicie sesión como raíz, con la contraseña: **password**.

Cambie a db2inst1 userid. El trabajar como raíz es como conducir sin un cinturón de seguridad. Por favor hable con su DBA local acerca de poner en ejecución su base de datos. Este artículo no cubrirá eso aquí. No intente instalar la base de datos dentro de la imagen virtual de Cloudera debido a que no hay suficiente espacio de disco libre.

La máquina virtual se conectará a la base de datos usando Sqoop, que requiere un controlador JDBC. Usted necesitará tener el controlador JDBC para su base de datos en la imagen virtual. Usted puede instalar el controlador Informix aquí.

El controlador DB2 está ubicado aquí: http://www.ibm.com/services/forms/preLogin.do?source=swg-idsdjs o http://www-01.ibm.com/support/docview.wss?rs=4020&uid=swg21385217.

La instalación del controlador Informix JDBC (recuerde, solo el controlador dentro de la imagen virtual, no la base de datos) se muestra en el Listado 1.

Listado 1. Instalación de controlador Informix JDBC

```
tar -xvf ../JDBC.3.70.JC5DE.tar
followed by
java -jar setup.jar
```

Nota: Seleccione un subdirectorio relativo a /home/cloudera para no requerir de permiso de raíz para la instalación.

El controlador DB2 JDBC está en formato zip, así que simplemente descomprímalo en el directorio de destino, como se muestra en el Listado 2.

Listado 2. DB2 JDBC driver install

```
mkdir db2jdbc
cd db2jdbc
unzip ../ibm_data_server_driver_for_jdbc_sqlj_v10.1.zip
```

Una rápida introducción a HDFS y MapReduce

Antes de iniciar a mover datos entre su base de datos relacional y Hadoop, usted necesita una rápida introducción a HDFS y a MapReduce. Hay muchos tutoriales estilo "hola mundo" para Hadoop, así que los ejemplos de aquí tienen la intención de darle solo suficientes antecedentes para que los ejercicios de base de datos tengan sentido para usted.

HDFS proporciona almacenamiento a lo largo de los nodos en su clúster. El primer paso al usar Hadoop es colocar los datos en HDFS. El código mostrado en el Listado 3 obtiene una copia de un libro por Mark Twain y un libro por James Fenimore Cooper y copia estos textos en HDFS.

Listado 3. Cargar Mark Twain y James Fenimore Cooper en HDFS

```
# install wget utility into the virtual image
sudo yum install wget

# use wget to download the Twain and Cooper's works
$ wget -U firefox http://www.gutenberg.org/cache/epub/76/pg76.txt
$ wget -U firefox http://www.gutenberg.org/cache/epub/3285/pg3285.txt

# load both into the HDFS file system
# first give the files better names
# DS for Deerslayer
# HF for  Huckleberry Finn
$ mv pg3285.txt DS.txt
$ mv pg76.txt HF.txt

# this next command will fail if the directory already exists
$ hadoop fs -mkdir /user/cloudera

# now put the text into the directory
$ hadoop fs -put HF.txt /user/cloudera


# way too much typing, create aliases for hadoop commands
$ alias hput="hadoop fs -put"
$ alias hcat="hadoop fs -cat"
$ alias hls="hadoop fs -ls"
# for CDH4
$ alias hrmr="hadoop fs -rm -r"
# for CDH3
$ alias hrmr="hadoop fs -rmr"

# load the other article
# but add some compression because we can

$ gzip DS.txt

# the  .  in the next command references the cloudera home directory
# in hdfs, /user/cloudera

$ hput DS.txt.gz .

# now take a look at the files we have in place
$ hls
Found 2 items
-rw-r--r-- 1 cloudera supergroup  459386 2012-08-08 19:34 /user/cloudera/DS.txt.gz
-rw-r--r-- 1 cloudera supergroup  597587 2012-08-08 19:35 /user/cloudera/HF.txt
```

Usted ahora tiene dos archivos en un directorio en HDFS. Por favor todavía no se emocione mucho. De verdad, en un único nodo y con solo cerca de 1 megabyte, estos es tan emocionante como ver cómo se seca la pintura. Pero si este fuera un clúster de 400 nodos y usted tuviera 5 petabytes en vivo, entonces usted no podría contener su emoción.

Muchos de los tutoriales Hadoop usan el ejemplo de conteo de palabras que se incluye en el ejemplo del archivo jar. Sucede que mucho del análisis involucra contar y agregar. El ejemplo en el in Listado 4 le muestra cómo invocar el contador de palabras.

Listado 4. Contando palabras en Twain y Cooper

```
# hadoop comes with some examples
# this next line uses the provided java implementation of a
# word count program

# for CDH4:
hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar wordcount HF.txt HF.out

# for CDH3:
hadoop jar /usr/lib/hadoop/hadoop-examples.jar wordcount HF.txt HF.out

# for CDH4:
hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar wordcount DS.txt.gz DS.out

# for CDH3:
hadoop jar /usr/lib/hadoop/hadoop-examples.jar wordcount  DS.txt.gz DS.out
```

El sufijo .gz de DS.txt.gz le pide a Hadoop lidiar con la descompresión como parte del procesamiento Map-Reduce. Cooper es un poco abundante, así que bien merece la compactación.

Hay bastantes secuencias de mensajes al ejecutar su trabajo de conteo de palabras. Hadoop está ávido de proporcionar muchos detalles acerca de los programas Mapping y Reducing que se ejecutan para usted. Las líneas críticas que usted desea buscar se muestran en el Listado 5, incluyendo un segundo listado de un trabajo fallido y cómo arreglar uno de los errores más comunes que encontrará al ejecutar MapReduce.

Listado 5. Mensajes de MapReduce - la "ruta feliz"

```
$ hadoop jar /usr/lib/hadoop/hadoop-examples.jar wordcount HF.txt HF.out
12/08/08 19:23:46 INFO input.FileInputFormat: Total input paths to process : 1
12/08/08 19:23:47 WARN snappy.LoadSnappy: Snappy native library is available
12/08/08 19:23:47 INFO util.NativeCodeLoader: Loaded the native-hadoop library
12/08/08 19:23:47 INFO snappy.LoadSnappy: Snappy native library loaded
12/08/08 19:23:47 INFO mapred.JobClient: Running job: job_201208081900_0002
12/08/08 19:23:48 INFO mapred.JobClient:  map 0% reduce 0%
12/08/08 19:23:54 INFO mapred.JobClient:  map 100% reduce 0%
12/08/08 19:24:01 INFO mapred.JobClient:  map 100% reduce 33%
12/08/08 19:24:03 INFO mapred.JobClient:  map 100% reduce 100%
12/08/08 19:24:04 INFO mapred.JobClient: Job complete: job_201208081900_0002
12/08/08 19:24:04 INFO mapred.JobClient: Counters: 26
12/08/08 19:24:04 INFO mapred.JobClient:   Job Counters
12/08/08 19:24:04 INFO mapred.JobClient:     Launched reduce tasks=1
12/08/08 19:24:04 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=5959
12/08/08 19:24:04 INFO mapred.JobClient:     Total time spent by all reduces...
12/08/08 19:24:04 INFO mapred.JobClient:     Total time spent by all maps waiting...
12/08/08 19:24:04 INFO mapred.JobClient:     Launched map tasks=1
12/08/08 19:24:04 INFO mapred.JobClient:     Data-local map tasks=1
12/08/08 19:24:04 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=9433
12/08/08 19:24:04 INFO mapred.JobClient:   FileSystemCounters
12/08/08 19:24:04 INFO mapred.JobClient:     FILE_BYTES_READ=192298
12/08/08 19:24:04 INFO mapred.JobClient:     HDFS_BYTES_READ=597700
12/08/08 19:24:04 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=498740
12/08/08 19:24:04 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=138218
12/08/08 19:24:04 INFO mapred.JobClient:   Map-Reduce Framework
12/08/08 19:24:04 INFO mapred.JobClient:     Map input records=11733
12/08/08 19:24:04 INFO mapred.JobClient:     Reduce shuffle bytes=192298
12/08/08 19:24:04 INFO mapred.JobClient:     Spilled Records=27676
12/08/08 19:24:04 INFO mapred.JobClient:     Map output bytes=1033012
12/08/08 19:24:04 INFO mapred.JobClient:     CPU time spent (ms)=2430
```

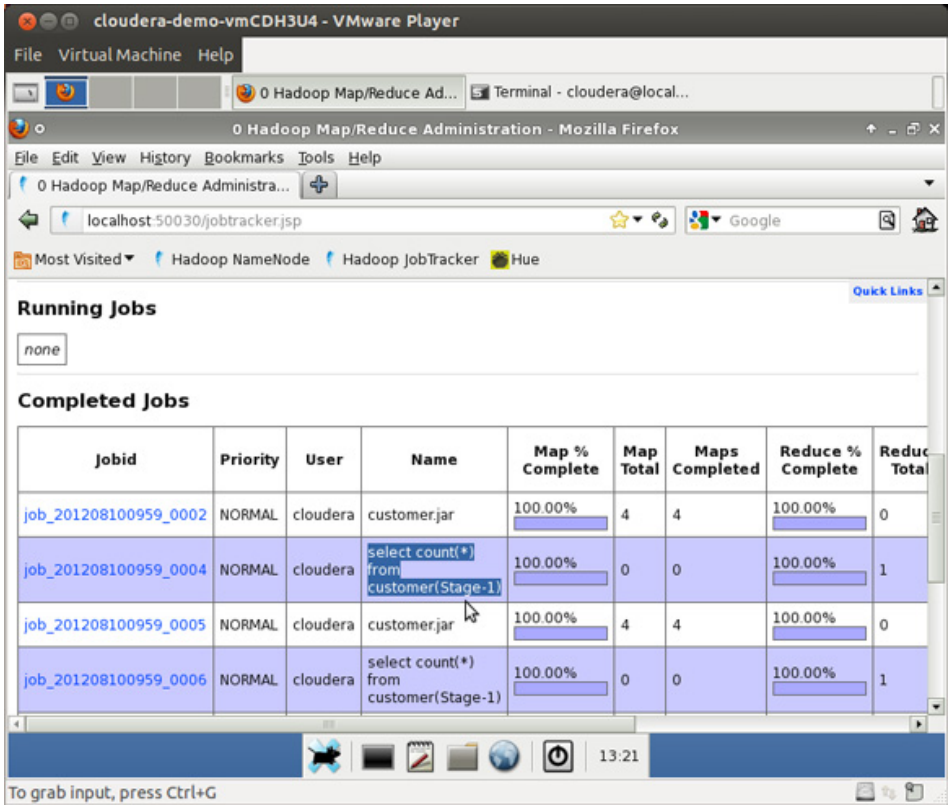
```
12/08/08 19:24:04 INFO mapred.JobClient:     Total committed heap usage (bytes)=183701504
12/08/08 19:24:04 INFO mapred.JobClient:     Combine input records=113365
12/08/08 19:24:04 INFO mapred.JobClient:     SPLIT_RAW_BYTES=113
12/08/08 19:24:04 INFO mapred.JobClient:     Reduce input records=13838
12/08/08 19:24:04 INFO mapred.JobClient:     Reduce input groups=13838
12/08/08 19:24:04 INFO mapred.JobClient:     Combine output records=13838
12/08/08 19:24:04 INFO mapred.JobClient:     Physical memory (bytes) snapshot=256479232
12/08/08 19:24:04 INFO mapred.JobClient:     Reduce output records=13838
12/08/08 19:24:04 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=1027047424
12/08/08 19:24:04 INFO mapred.JobClient:     Map output records=113365
```

¿Qué significan todos estos mensajes? Hadoop ha efectuado mucho trabajo y está intentando comunicárselo a usted, incluyendo lo siguiente.

- Verificó si el archivo de entrada existe.
- Verificó su el directorio de salida existe, y si existe, aborta el trabajo. No hay nada peor que horas de cómputo duplicado debido a un simple error de digitación.
- Distribuyó el archivo Java jar hacia todos los nodos responsables por realizar el trabajo. En este caso, este es solo un nodo.
- Ejecutó la fase mapper del trabajo. Típicamente esto analiza el archivo de entrada y emite un par de valor clave. Observe que la clave y valor pueden ser objetos.
- Ejecutó la fase de ordenar, que ordena la salida del mapper con base en la clave.
- Ejecutó la fase de reducción, típicamente esto resume la transmisión de clave-valor y escribe salida hacia HDFS.
- Creó muchas métricas en el transcurso.

La Figura 6 muestra una página de ejemplo de las métricas de trabajo Hadoop después de ejecutar el ejercicio Hive.

Figura 6. Muestra de página web de Hadoop



¿Qué hizo el trabajo y dónde está la salida? Ambas son buenas preguntas, y se muestran en el Listado 6.

Listado 6. Salida de Map-Reduce

```
# way too much typing, create aliases for hadoop commands
$ alias hput="hadoop fs -put"
$ alias hcat="hadoop fs -cat"
$ alias hls="hadoop fs -ls"
$ alias hrmr="hadoop fs -rmr"

# first list the output directory
$ hls /user/cloudera/HF.out
Found 3 items
-rw-r--r-- 1 cloudera supergroup 0 2012-08-08 19:38 /user/cloudera/HF.out/_SUCCESS
drwxr-xr-x - cloudera supergroup 0 2012-08-08 19:38 /user/cloudera/HF.out/_logs
-rw-r--r-- 1 cl... sup... 138218 2012-08-08 19:38 /user/cloudera/HF.out/part-r-00000

# now cat the file and pipe it to  the  less command
$ hcat /user/cloudera/HF.out/part-r-00000 | less

# here are a few lines from the file, the word elephants only got used twice
elder,  1
eldest  1
elect   1
elected 1
electronic      27
electronically  1
electronically, 1
elegant 1
elegant!--'deed 1
elegant,        1
```

```
elephants       2
```

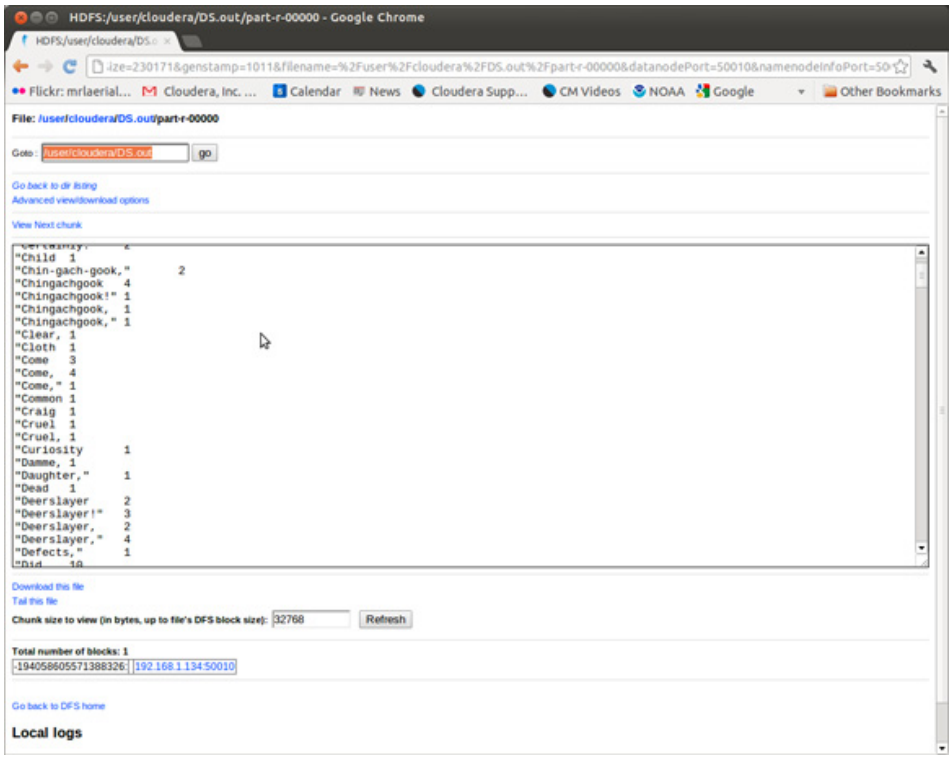
En la eventualidad de que usted ejecute el mismo trabajo dos veces y olvide suprimir el directorio de salida, usted recibirá los mensajes de error mostrados en el Listado 7. El arreglo de este error es tan sencillo como suprimir el directorio.

Listado 7. Mensajes MapReduce - falla debida a que la salida ya existe en HDFS

```
# way too much typing, create aliases for hadoop commands
$ alias hput="hadoop fs -put"
$ alias hcat="hadoop fs -cat"
$ alias hls="hadoop fs -ls"
$ alias hrmr="hadoop fs -rmr"

$ hadoop jar /usr/lib/hadoop/hadoop-examples.jar wordcount HF.txt HF.out
12/08/08 19:26:23 INFO mapred.JobClient:
Cleaning up the staging area hdfs://0.0.0.0/var/l...
12/08/08 19:26:23 ERROR security.UserGroupInformation: PriviledgedActionException
as:cloudera (auth:SIMPLE)
cause:org.apache.hadoop.mapred.FileAlreadyExistsException:
Output directory HF.out already exists
org.apache.hadoop.mapred.FileAlreadyExistsException:
Output directory HF.out already exists
at org.apache.hadoop.mapreduce.lib.output.FileOutputFormat.
checkOutputSpecs(FileOutputFormat.java:132)
at org.apache.hadoop.mapred.JobClient$2.run(JobClient.java:872)
at org.apache.hadoop.mapred.JobClient$2.run(JobClient.java:833)

.... lines deleted

# the simple fix is to remove the existing output directory

$ hrmr HF.out

# now you can re-run the job successfully

# if you run short of space and the namenode enters safemode
# clean up some file space and then

$ hadoop dfsadmin -safemode leave
```

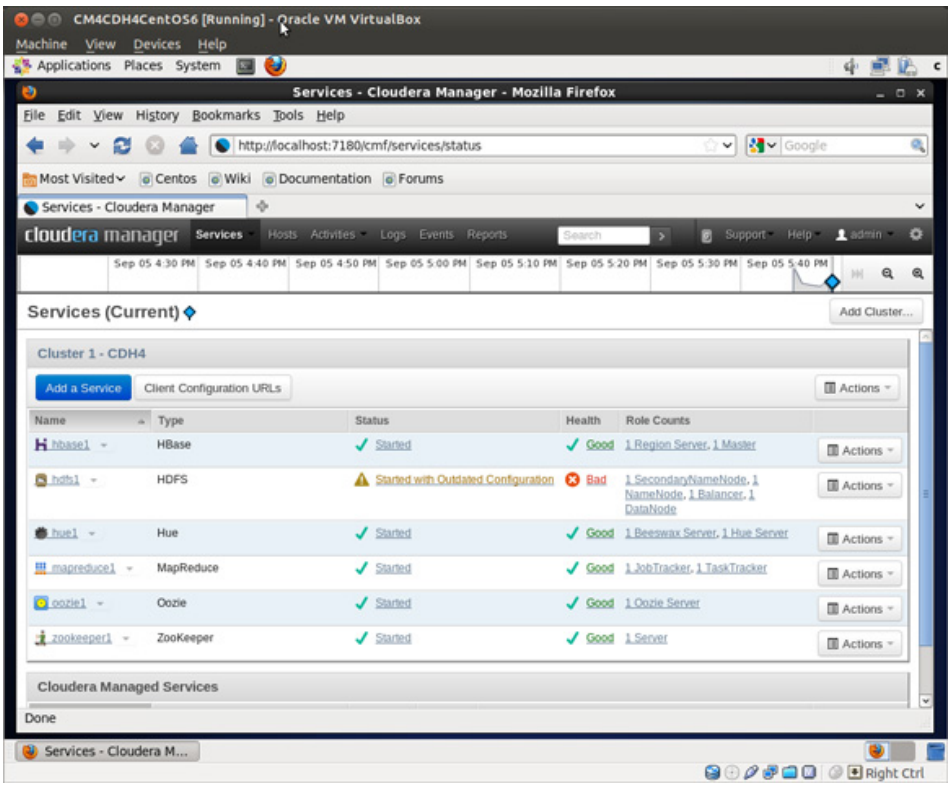
Hadoop incluye una interfaz de usuario para inspeccionar el estado de HDFS. La Figura 7 muestra la salida del trabajo de conteo de palabras.

Figura 7. Explorando HDFS con un navegador



Una consola más sofisticada está disponible gratuitamente en el sitio web de Cloudera. Proporciona diversas posibilidades que van más allá de las interfaces web Hadoop estándares. Observe que el estado de salud de HDFS en la Figura 8 se muestra como **Bad**.

Figura 8. Servicios Hadoop gestionados por Cloudera Manager



¿Por qué mala? Debido a que un una única máquina virtual, HDFS no puede hacer tres copias de bloques de datos. Cuando los bloques están sub-replicados, entonces hay riesgo de pérdida de datos, así que la salud del sistema es mala. Lo bueno es que usted no está intentando ejecutar trabajos de producción Hadoop en un único nodo.

Usted no está limitado a Java para sus trabajos MapReduce. Este último ejemplo de MapReduce usa Hadoop Streaming para dar soporte a un correlacionador escrito en Python y un reductor usando AWK. No, ¡usted no necesita ser un gurú Java para escribir Map-Reduce!

Mark Twain no era un gran admirador de Cooper. En este caso de uso, Hadoop proporcionará algunas críticas literarias sencillas comparando entre Twain y Cooper. La prueba Flesch–Kincaid calcula el nivel de lectura de un texto particular. Uno de los factores de este análisis el la longitud promedio de la sentencia. El análisis de sentencias resulta ser más complicado que solo buscar el carácter de punto. El paquete openNLP y el paquete Python NLTK tienen excelentes analizadores de sentencias. Para obtener simplicidad, el ejemplo mostrado en el Listado 8 usará longitud de palabras como un sustituto para el número de sílabas en una palabra. Si desea llevar esto al próximo nivel, implemente la prueba Flesch–Kincaid en MapReduce, busque en la web, y calcule niveles de lectura para sus sitios de noticias favoritos.

Listado 8. Una crítica literaria de correlacionador basado en Python

```
# here is the mapper we'll connect to the streaming hadoop interface

# the mapper is reading the text in the file - not really appreciating Twain's humor
#
```

```
# modified from
# http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/
$ cat mapper.py
#!/usr/bin/env python
import sys

# read stdin
for linein in sys.stdin:
# strip blanks
linein = linein.strip()
# split into words
mywords = linein.split()
# loop on mywords, output the length of each word
for word in mywords:
# the reducer just cares about the first column,
# normally there is a key - value pair
print '%s %s' % (len(word), 0)
```

La salida del correlacionador para la palabra "Twain", sería: 5 0. Las longitudes numéricas de palabras son ordenadas y presentadas al reductor ordenadamente. En los ejemplos mostrados en los Listados 9 y 10, no se requiere ordenas los datos para obtener la salida correcta, pero el proceso de orden está incorporado en la infra estructura de MapReduce y de cualquier manera se dará.

Listado 9. Un reductor AWK para crítica literaria

```
# the awk code is modified from http://www.commandlinefu.com

# awk is calculating
#  NR - the number of words in total
#  sum/NR - the average word length
# sqrt(mean2/NR) - the standard deviation

$ cat statsreducer.awk
awk '{delta = $1 - avg; avg += delta / NR; \
mean2 += delta * ($1 - avg); sum=$1+sum } \
END { print NR, sum/NR, sqrt(mean2 / NR); }'
```

Listado 10. Ejecutando un correlacionador Python y un reductor AWK con Hadoop Streaming

```
# test locally

# because we're using Hadoop Streaming, we can test the
# mapper and reducer with simple pipes

# the "sort" phase is a reminder the keys are sorted
# before presentation to the reducer
#in this example it doesn't matter what order the
# word length values are presented for calculating the std deviation

$ zcat ../DS.txt.gz  | ./mapper.py | sort | ./statsreducer.awk
215107 4.56068 2.50734

# now run in hadoop with streaming

# CDH4
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \
-input HF.txt -output HFstats -file ./mapper.py -file \
./statsreducer.awk -mapper ./mapper.py -reducer ./statsreducer.awk

# CDH3
$ hadoop jar /usr/lib/hadoop-0.20/contrib/streaming/hadoop-streaming-0.20.2-cdh3u4.jar \
```

```
-input HF.txt -output HFstats -file ./mapper.py -file ./statsreducer.awk \
-mapper ./mapper.py -reducer ./statsreducer.awk

$ hls HFstats
Found 3 items
-rw-r--r--   1 cloudera supergroup   0 2012-08-12 15:38 /user/cloudera/HFstats/_SUCCESS
drwxr-xr-x   - cloudera supergroup   0 2012-08-12 15:37 /user/cloudera/HFstats/_logs
-rw-r--r--   1 cloudera ...  24 2012-08-12 15:37 /user/cloudera/HFstats/part-00000

$ hcat /user/cloudera/HFstats/part-00000
113365 4.11227 2.17086

# now for cooper

$ hadoop jar /usr/lib/hadoop-0.20/contrib/streaming/hadoop-streaming-0.20.2-cdh3u4.jar \
-input DS.txt.gz -output DSstats -file ./mapper.py -file ./statsreducer.awk \
-mapper ./mapper.py -reducer ./statsreducer.awk

$ hcat /user/cloudera/DSstats/part-00000
215107 4.56068 2.50734
```

Los admiradores de Mark Twain pueden estar tranquilos sabiendo que Hadoop descubrió que Cooper usa palabras más largas, y con una "desviación estándar alarmante" (es una broma). Eso desde luego asume que las palabras más cortas son mejores. Avancemos, lo siguiente es escribir los datos en HDFS para Informix y DB2.

Usando Sqoop para escribir datos de HDFS hacia Informix, DB2, o MySQL vía JDBC

Sqoop Apache Project es una utilidad de movimiento de datos de código abierto basada en JDBC de Hadoop a base de datos. Sqoop se creó originalmente en un hackathon en Cloudera y luego se transfirió a código abierto.

Mover datos de HDFS hacia una base de datos relacional es un caso de uso común. HDFS y Map-Reduce son magníficos para realizar el trabajo pesado. Para consultas sencillas o almacenamiento de back-end para un sitio web, agarrar la salida de Map-Reduce en un almacenamiento relacional representa un buen patrón de diseño. Usted puede evitar volver a ejecutar el conteo de palabras de Map-Reduce simplemente haciendo el Sqooping de los resultados en Informix y DB2. Usted ha generado datos acerca de Twain y Cooper, ahora trasladémonos hacia una base de datos, como se muestra en el Listado 11.

Listado 11. Configuración de controlador JDBC

```
#Sqoop needs access to the JDBC driver for every
# database that it will access

# please copy the driver for each database you plan to use for these exercises
# the MySQL database and driver are already installed in the virtual image
# but you still need to copy the driver to the sqoop/lib directory

#one time copy of jdbc driver to sqoop lib directory
$ sudo cp Informix_JDBC_Driver/lib/ifxjdbc*.jar /usr/lib/sqoop/lib/
$ sudo cp db2jdbc/db2jcc*.jar /usr/lib/sqoop/lib/
$ sudo cp /usr/lib/hive/lib/mysql-connector-java-5.1.15-bin.jar /usr/lib/sqoop/lib/
```

Los ejemplos mostrados en los Listados 12 a 15 se presentan para cada base de datos. Por favor vaya al ejemplo de interés para usted, incluyendo Informix, DB2, o MySQL. Los políglotas de

base de datos pueden divertise haciendo cada ejemplo. Si su base de datos de preferencia no se incluye aquí, no será un gran desafío hacer que estas muestras funcionen en otro lugar.

Listado 12. Usuarios de Informix: Sqoop escribiendo los resultados del conteo de palabras hacia Informix

```
# create a target table to put the data
# fire up dbaccess and use this sql
# create table wordcount ( word char(36) primary key, n int);

# now run the sqoop command
# this is best put in a shell script to help avoid typos...

$ sqoop export -D sqoop.export.records.per.statement=1 \
--fields-terminated-by '\t' --driver com.informix.jdbc.IfxDriver \
--connect \
"jdbc:informix-sqli://myhost:54321/stores_demo:informixserver=i7;user=me;password=mypw" \
--table wordcount --export-dir /user/cloudera/HF.out
```

Listado 13. Usuarios de Informix: Sqoop escribiendo los resultados del conteo de palabras hacia Informix

```
12/08/08 21:39:42 INFO manager.SqlManager: Using default fetchSize of 1000
12/08/08 21:39:42 INFO tool.CodeGenTool: Beginning code generation
12/08/08 21:39:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.*
FROM wordcount AS t WHERE 1=0
12/08/08 21:39:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.*
FROM wordcount AS t WHERE 1=0
12/08/08 21:39:43 INFO orm.CompilationManager: HADOOP_HOME is /usr/lib/hadoop
12/08/08 21:39:43 INFO orm.CompilationManager: Found hadoop core jar at:
/usr/lib/hadoop/hadoop-0.20.2-cdh3u4-core.jar
12/08/08 21:39:45 INFO orm.CompilationManager: Writing jar file:
/tmp/sqoop-cloudera/compile/248b77c05740f863a15e0136accf32cf/wordcount.jar
12/08/08 21:39:45 INFO mapreduce.ExportJobBase: Beginning export of wordcount
12/08/08 21:39:45 INFO manager.SqlManager: Executing SQL statement: SELECT t.*
FROM wordcount AS t WHERE 1=0
12/08/08 21:39:46 INFO input.FileInputFormat: Total input paths to process : 1
12/08/08 21:39:46 INFO input.FileInputFormat: Total input paths to process : 1
12/08/08 21:39:46 INFO mapred.JobClient: Running job: job_201208081900_0012
12/08/08 21:39:47 INFO mapred.JobClient:  map 0% reduce 0%
12/08/08 21:39:58 INFO mapred.JobClient:  map 38% reduce 0%
12/08/08 21:40:00 INFO mapred.JobClient:  map 64% reduce 0%
12/08/08 21:40:04 INFO mapred.JobClient:  map 82% reduce 0%
12/08/08 21:40:07 INFO mapred.JobClient:  map 98% reduce 0%
12/08/08 21:40:09 INFO mapred.JobClient: Task Id :
attempt_201208081900_0012_m_000000_0, Status : FAILED
java.io.IOException: java.sql.SQLException:
    Encoding or code set not supported.
at ...SqlRecordWriter.close(AsyncSqlRecordWriter.java:187)
at ...$NewDirectOutputCollector.close(MapTask.java:540)
at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:649)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java:323)
at org.apache.hadoop.mapred.Child$4.run(Child.java:270)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:396)
at ....doAs(UserGroupInformation.java:1177)
at org.apache.hadoop.mapred.Child.main(Child.java:264)
Caused by: java.sql.SQLException: Encoding or code set not supported.
at com.informix.util.IfxErrMsg.getSQLException(IfxErrMsg.java:413)
at com.informix.jdbc.IfxChar.toIfx(IfxChar.java:135)
at com.informix.jdbc.IfxSqli.a(IfxSqli.java:1304)
at com.informix.jdbc.IfxSqli.d(IfxSqli.java:1605)
at com.informix.jdbc.IfxS
12/08/08 21:40:11 INFO mapred.JobClient:  map 0% reduce 0%
```

```
12/08/08 21:40:15 INFO mapred.JobClient: Task Id :
attempt_201208081900_0012_m_000000_1, Status : FAILED
java.io.IOException: java.sql.SQLException:
    Unique constraint (informix.u169_821) violated.
at .mapreduce.AsyncSqlRecordWriter.write(AsyncSqlRecordWriter.java:223)
at .mapreduce.AsyncSqlRecordWriter.write(AsyncSqlRecordWriter.java:49)
at .mapred.MapTask$NewDirectOutputCollector.write(MapTask.java:531)
at .mapreduce.TaskInputOutputContext.write(TaskInputOutputContext.java:80)
at com.cloudera.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:82)
at com.cloudera.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:40)
at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:144)
at .mapreduce.AutoProgressMapper.run(AutoProgressMapper.java:189)
at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:647)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java:323)
at org.apache.hadoop.mapred.Child$4.run(Child.java:270)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.a
12/08/08 21:40:20 INFO mapred.JobClient:
Task Id : attempt_201208081900_0012_m_000000_2, Status : FAILED
java.sql.SQLException: Unique constraint (informix.u169_821) violated.
at .mapreduce.AsyncSqlRecordWriter.write(AsyncSqlRecordWriter.java:223)
at .mapreduce.AsyncSqlRecordWriter.write(AsyncSqlRecordWriter.java:49)
at .mapred.MapTask$NewDirectOutputCollector.write(MapTask.java:531)
at .mapreduce.TaskInputOutputContext.write(TaskInputOutputContext.java:80)
at com.cloudera.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:82)
at com.cloudera.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:40)
at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:144)
at .mapreduce.AutoProgressMapper.run(AutoProgressMapper.java:189)
at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:647)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java:323)
at org.apache.hadoop.mapred.Child$4.run(Child.java:270)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.a
12/08/08 21:40:27 INFO mapred.JobClient: Job complete: job_201208081900_0012
12/08/08 21:40:27 INFO mapred.JobClient: Counters: 7
12/08/08 21:40:27 INFO mapred.JobClient:   Job Counters
12/08/08 21:40:27 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=38479
12/08/08 21:40:27 INFO mapred.JobClient:
Total time spent by all reduces waiting after reserving slots (ms)=0
12/08/08 21:40:27 INFO mapred.JobClient:
Total time spent by all maps waiting after reserving slots (ms)=0
12/08/08 21:40:27 INFO mapred.JobClient:     Launched map tasks=4
12/08/08 21:40:27 INFO mapred.JobClient:     Data-local map tasks=4
12/08/08 21:40:27 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
12/08/08 21:40:27 INFO mapred.JobClient:     Failed map tasks=1
12/08/08 21:40:27 INFO mapreduce.ExportJobBase:
Transferred 0 bytes in 41.5758 seconds (0 bytes/sec)
12/08/08 21:40:27 INFO mapreduce.ExportJobBase: Exported 0 records.
12/08/08 21:40:27 ERROR tool.ExportTool: Error during export: Export job failed!

# despite the errors above, rows are inserted into the wordcount table
# one row is missing
# the retry and duplicate key exception are most likely related, but
# troubleshooting will be saved for a later article

# check how we did
# nothing like a "here document" shell script

$ dbaccess stores_demo - <<eoj
> select count(*) from wordcount;
> eoj

Database selected.
(count(*))
13837
1 row(s) retrieved.
Database closed.
```

Listado 14. Usuarios de DB2: Sqoop escribiendo los resultados del conteo de palabras hacia DB2

```
# here is the db2 syntax
# create a destination table for db2
#
#db2 => connect to sample
#
#   Database Connection Information
#
# Database server        = DB2/LINUXX8664 10.1.0
# SQL authorization ID   = DB2INST1
# Local database alias   = SAMPLE
#
#db2 => create table wordcount ( word char(36) not null primary key , n int)
#DB20000I  The SQL command completed successfully.
#

sqoop export -D sqoop.export.records.per.statement=1 \
--fields-terminated-by '\t' \
--driver com.ibm.db2.jcc.DB2Driver \
--connect "jdbc:db2://192.168.1.131:50001/sample"  \
--username db2inst1 --password db2inst1 \
--table wordcount --export-dir /user/cloudera/HF.out

12/08/09 12:32:59 WARN tool.BaseSqoopTool: Setting your password on the
command-line is insecure. Consider using -P instead.
12/08/09 12:32:59 INFO manager.SqlManager: Using default fetchSize of 1000
12/08/09 12:32:59 INFO tool.CodeGenTool: Beginning code generation
12/08/09 12:32:59 INFO manager.SqlManager: Executing SQL statement:
SELECT t.* FROM wordcount AS t WHERE 1=0
12/08/09 12:32:59 INFO manager.SqlManager: Executing SQL statement:
SELECT t.* FROM wordcount AS t WHERE 1=0
12/08/09 12:32:59 INFO orm.CompilationManager: HADOOP_HOME is /usr/lib/hadoop
12/08/09 12:32:59 INFO orm.CompilationManager: Found hadoop core jar
at: /usr/lib/hadoop/hadoop-0.20.2-cdh3u4-core.jar
12/08/09 12:33:00 INFO orm.CompilationManager: Writing jar
file: /tmp/sqoop-cloudera/compile/5532984df6e28e5a45884a21bab245ba/wordcount.jar
12/08/09 12:33:00 INFO mapreduce.ExportJobBase: Beginning export of wordcount
12/08/09 12:33:01 INFO manager.SqlManager: Executing SQL statement:
SELECT t.* FROM wordcount AS t WHERE 1=0
12/08/09 12:33:02 INFO input.FileInputFormat: Total input paths to process : 1
12/08/09 12:33:02 INFO input.FileInputFormat: Total input paths to process : 1
12/08/09 12:33:02 INFO mapred.JobClient: Running job: job_201208091208_0002
12/08/09 12:33:03 INFO mapred.JobClient:  map 0% reduce 0%
12/08/09 12:33:14 INFO mapred.JobClient:  map 24% reduce 0%
12/08/09 12:33:17 INFO mapred.JobClient:  map 44% reduce 0%
12/08/09 12:33:20 INFO mapred.JobClient:  map 67% reduce 0%
12/08/09 12:33:23 INFO mapred.JobClient:  map 86% reduce 0%
12/08/09 12:33:24 INFO mapred.JobClient:  map 100% reduce 0%
12/08/09 12:33:25 INFO mapred.JobClient: Job complete: job_201208091208_0002
12/08/09 12:33:25 INFO mapred.JobClient: Counters: 16
12/08/09 12:33:25 INFO mapred.JobClient:   Job Counters
12/08/09 12:33:25 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=21648
12/08/09 12:33:25 INFO mapred.JobClient:     Total time spent by all
reduces waiting after reserving slots (ms)=0
12/08/09 12:33:25 INFO mapred.JobClient:     Total time spent by all
maps waiting after reserving slots (ms)=0
12/08/09 12:33:25 INFO mapred.JobClient:     Launched map tasks=1
12/08/09 12:33:25 INFO mapred.JobClient:     Data-local map tasks=1
12/08/09 12:33:25 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
12/08/09 12:33:25 INFO mapred.JobClient:   FileSystemCounters
12/08/09 12:33:25 INFO mapred.JobClient:     HDFS_BYTES_READ=138350
12/08/09 12:33:25 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=69425
12/08/09 12:33:25 INFO mapred.JobClient:   Map-Reduce Framework
12/08/09 12:33:25 INFO mapred.JobClient:     Map input records=13838
12/08/09 12:33:25 INFO mapred.JobClient:     Physical memory (bytes) snapshot=105148416
```

```
12/08/09 12:33:25 INFO mapred.JobClient:     Spilled Records=0
12/08/09 12:33:25 INFO mapred.JobClient:     CPU time spent (ms)=9250
12/08/09 12:33:25 INFO mapred.JobClient:     Total committed heap usage (bytes)=42008576
12/08/09 12:33:25 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=596447232
12/08/09 12:33:25 INFO mapred.JobClient:     Map output records=13838
12/08/09 12:33:25 INFO mapred.JobClient:     SPLIT_RAW_BYTES=126
12/08/09 12:33:25 INFO mapreduce.ExportJobBase: Transferred 135.1074 KB
in 24.4977 seconds (5.5151 KB/sec)
12/08/09 12:33:25 INFO mapreduce.ExportJobBase: Exported 13838 records.

# check on the results...
#
#db2 => select count(*) from wordcount
#
#1
#-----------
#      13838
#
#  1 record(s) selected.
#
#
```

Listado 15. Usuarios de MySQL: Sqoop escribiendo los resultados del conteo de palabras hacia MySQL

```
# if you don't have Informix or DB2 you can still do this example
# mysql - it is already installed in the VM, here is how to access

# one time copy of the JDBC driver

sudo cp /usr/lib/hive/lib/mysql-connector-java-5.1.15-bin.jar /usr/lib/sqoop/lib/

# now create the database and table

$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.0.95 Source distribution

Copyright (c) 2000, 2011, Oracle y/o sus afiliadas. Todos los derechos reservados.

Oracle es una marca registrada de Oracle Corporation y/o sus afiliadas. Otros nombres pueden ser marcas
 registradas de sus respectivos dueños.

Escriba 'help;' o '\h' para obtener ayuda. Escriba '\c' para limpiar la sentencia de entrada  actual.

mysql> create database mydemo;
Query OK, 1 row affected (0.00 sec)

mysql> use mydemo
Database changed
mysql> create table wordcount ( word char(36) not null primary key, n int);
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye

# now export

$ sqoop export --connect jdbc:mysql://localhost/mydemo \
--table wordcount --export-dir /user/cloudera/HF.out \
--fields-terminated-by '\t' --username root
```

Importando datos hacia HDFS desde Informix y DB2 con Sqoop

También se pueden insertar datos en Hadoop HDFS con Sqoop. La funcionalidad bidireccional es controlada vía el parámetro de importación.

Las bases de datos de muestra que vienen con ambos productos tienen algunos datasets sencillos que usted puede usar para este propósito. El Listado 16 muestra la sintaxis y resultados para la realización de Sqooping en cada servidor.

Para usuarios de MySQL, por favor adapten la sintaxis de los ejemplos de Informix o de DB2 que siguen.

Listado 16. Importación de Sqoop de una base de datos de muestra de Informix hacia HDFS

```
$ sqoop import --driver com.informix.jdbc.IfxDriver \
--connect \
"jdbc:informix-sqli://192.168.1.143:54321/stores_demo:informixserver=ifx117" \
--table orders \
--username informix --password useyours

12/08/09 14:39:18 WARN tool.BaseSqoopTool: Setting your password on the command-line
is insecure. Consider using -P instead.
12/08/09 14:39:18 INFO manager.SqlManager: Using default fetchSize of 1000
12/08/09 14:39:18 INFO tool.CodeGenTool: Beginning code generation
12/08/09 14:39:19 INFO manager.SqlManager: Executing SQL statement:
SELECT t.* FROM orders AS t WHERE 1=0
12/08/09 14:39:19 INFO manager.SqlManager: Executing SQL statement:
SELECT t.* FROM orders AS t WHERE 1=0
12/08/09 14:39:19 INFO orm.CompilationManager: HADOOP_HOME is /usr/lib/hadoop
12/08/09 14:39:19 INFO orm.CompilationManager: Found hadoop core jar
at: /usr/lib/hadoop/hadoop-0.20.2-cdh3u4-core.jar
12/08/09 14:39:21 INFO orm.CompilationManager: Writing jar
file: /tmp/sqoop-cloudera/compile/0b59eec7007d3cff1fc0ae446ced3637/orders.jar
12/08/09 14:39:21 INFO mapreduce.ImportJobBase: Beginning import of orders
12/08/09 14:39:21 INFO manager.SqlManager: Executing SQL statement:
SELECT t.* FROM orders AS t WHERE 1=0
12/08/09 14:39:22 INFO db.DataDrivenDBInputFormat: BoundingValsQuery:
SELECT MIN(order_num), MAX(order_num) FROM orders
12/08/09 14:39:22 INFO mapred.JobClient: Running job: job_201208091208_0003
12/08/09 14:39:23 INFO mapred.JobClient:  map 0% reduce 0%
12/08/09 14:39:31 INFO mapred.JobClient:  map 25% reduce 0%
12/08/09 14:39:32 INFO mapred.JobClient:  map 50% reduce 0%
12/08/09 14:39:36 INFO mapred.JobClient:  map 100% reduce 0%
12/08/09 14:39:37 INFO mapred.JobClient: Job complete: job_201208091208_0003
12/08/09 14:39:37 INFO mapred.JobClient: Counters: 16
12/08/09 14:39:37 INFO mapred.JobClient:   Job Counters
12/08/09 14:39:37 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=22529
12/08/09 14:39:37 INFO mapred.JobClient:     Total time spent by all reduces
waiting after reserving slots (ms)=0
12/08/09 14:39:37 INFO mapred.JobClient:     Total time spent by all maps
waiting after reserving slots (ms)=0
12/08/09 14:39:37 INFO mapred.JobClient:     Launched map tasks=4
12/08/09 14:39:37 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
12/08/09 14:39:37 INFO mapred.JobClient:   FileSystemCounters
12/08/09 14:39:37 INFO mapred.JobClient:     HDFS_BYTES_READ=457
12/08/09 14:39:37 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=278928
12/08/09 14:39:37 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=2368
12/08/09 14:39:37 INFO mapred.JobClient:   Map-Reduce Framework
12/08/09 14:39:37 INFO mapred.JobClient:     Map input records=23
12/08/09 14:39:37 INFO mapred.JobClient:     Physical memory (bytes) snapshot=291364864
12/08/09 14:39:37 INFO mapred.JobClient:     Spilled Records=0
12/08/09 14:39:37 INFO mapred.JobClient:     CPU time spent (ms)=1610
```

```
12/08/09 14:39:37 INFO mapred.JobClient:     Total committed heap usage (bytes)=168034304
12/08/09 14:39:37 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=2074587136
12/08/09 14:39:37 INFO mapred.JobClient:     Map output records=23
12/08/09 14:39:37 INFO mapred.JobClient:     SPLIT_RAW_BYTES=457
12/08/09 14:39:37 INFO mapreduce.ImportJobBase: Transferred 2.3125 KB in 16.7045
seconds (141.7585 bytes/sec)
12/08/09 14:39:37 INFO mapreduce.ImportJobBase: Retrieved 23 records.

# now look at the results

$ hls
Found 4 items
-rw-r--r--   1 cloudera supergroup     459386 2012-08-08 19:34 /user/cloudera/DS.txt.gz
drwxr-xr-x   - cloudera supergroup          0 2012-08-08 19:38 /user/cloudera/HF.out
-rw-r--r--   1 cloudera supergroup     597587 2012-08-08 19:35 /user/cloudera/HF.txt
drwxr-xr-x   - cloudera supergroup          0 2012-08-09 14:39 /user/cloudera/orders
$ hls orders
Found 6 items
-rw-r--r--   1 cloudera supergroup 0 2012-08-09 14:39 /user/cloudera/orders/_SUCCESS
drwxr-xr-x   - cloudera supergroup 0 2012-08-09 14:39 /user/cloudera/orders/_logs
-rw-r--r--   1 cloudera ...roup 630 2012-08-09 14:39 /user/cloudera/orders/part-m-00000
-rw-r--r--   1 cloudera supergroup
564 2012-08-09 14:39 /user/cloudera/orders/part-m-00001
-rw-r--r--   1 cloudera supergroup
527 2012-08-09 14:39 /user/cloudera/orders/part-m-00002
-rw-r--r--   1 cloudera supergroup
647 2012-08-09 14:39 /user/cloudera/orders/part-m-00003

# wow  there are four files part-m-0000x
# look inside one

# some of the lines are edited to fit on the screen
$ hcat /user/cloudera/orders/part-m-00002
1013,2008-06-22,104,express ,n,B77930    ,2008-07-10,60.80,12.20,2008-07-31
1014,2008-06-25,106,ring bell,  ,n,8052      ,2008-07-03,40.60,12.30,2008-07-10
1015,2008-06-27,110,        ,n,MA003     ,2008-07-16,20.60,6.30,2008-08-31
1016,2008-06-29,119, St.          ,n,PC6782    ,2008-07-12,35.00,11.80,null
1017,2008-07-09,120,use                ,n,DM354331  ,2008-07-13,60.00,18.00,null
```

¿Por qué hay cuatro diferentes archivos conteniendo solo parte de los datos? Sqoop es una utilidad altamente paralelizada. Si un clúster de 4000 nodos ejecutando Sqoop realizó una importación de regulador completo de una base de datos, las 4000 conexiones se verían mucho como un ataque de denegación de servicio contra la base de datos. El límite de conexión predeterminado de Sqoop es de cuatro conexiones JDBC. Ca da conexión genera un archivo de datos en HDFS. Por lo tanto los cuatro archivos. No hay que preocuparse, usted verá cómo funciona Hadoop a lo largo de estos archivos sin ninguna dificultad.

El siguiente paso es importar una tabla DB2. Como se muestra en el Listado 17, al especificar la opción **-m 1** , se puede importar una tabla sin una clave primaria, y el resultado es un único archivo.

Listado 17. Importación Sqoop de una base de datos DB2 de muestra hacia HDFS

```
# very much the same as above, just a different jdbc connection
# and different table name

sqoop import --driver com.ibm.db2.jcc.DB2Driver \
--connect "jdbc:db2://192.168.1.131:50001/sample"  \
--table staff --username db2inst1  \
```

```
--password db2inst1 -m 1

# Here is another example
# in this case set the sqoop default schema to be different from
# the user login schema

sqoop import --driver com.ibm.db2.jcc.DB2Driver \
--connect "jdbc:db2://192.168.1.3:50001/sample:currentSchema=DB2INST1;" \
--table helloworld \
--target-dir "/user/cloudera/sqoopin2" \
--username marty \
-P -m 1

# the the schema name is CASE SENSITIVE
# the -P option prompts for a password that will not be visible in
# a "ps" listing
```

Usando Hive: Uniendo datos de Informix y DB2

Existe un uso de caso interesante para unir datos de Informix hacia DB2. No muy emocionante para dos tablas triviales, pero es una inmensa ganancia para múltiples terabytes o petabytes de datos.

Existen dos abordajes fundamentales para unir diferente fuentes de datos. Dejando los datos inmóviles y usando tecnología de federación versus mover los datos hacia un almacenamiento único para realizar la unión. La economía y rendimiento de Hadoop hace que mover los datos hacia HDFS y realizar el trabajo pesado con MapReduce sea una opción fácil. Las limitaciones de ancho de banda de la red crean una barrera fundamental si se trata de unir datos inmóviles con tecnología estilo federación. P ara obtener más información acerca de federación, por favor vea Recursos .

Hive proporciona un subconjunto de SQL para operar en un clúster. No proporciona semántica de transacciones. No es un reemplazo para Informix o DB2. Si usted tiene trabajo pesado bajo la forma de uniones de tabla, incluso si usted tiene algunas tablas menores pero necesita realizar productos Cartesian no agradables, Hadoop es la herramienta por la que debe optar.

Para usar el lenguaje de consulta Hive, se requiere de un subconjunto de SQL llamado tabla de metadatos Hiveql. Usted puede definir los metadatos contra archivos existentes en HDFS. Sqoop proporciona un atajo conveniente con la opción create-hive-table.

Los usuarios de MySQL deben sentirse en libertad de adaptar los ejemplos mostrados en el Listado 18. Un ejercicio interesante sería unir MySQL, o cualquier otra tabla de base de datos relacional, a grandes hojas de cálculo.

Listado 18. Uniendo la tabla informix.customer a la tabla db2.staff

```
# import the customer table into Hive
$ sqoop import --driver com.informix.jdbc.IfxDriver  \
--connect \
"jdbc:informix-sqli://myhost:54321/stores_demo:informixserver=ifx;user=me;password=you"  \
--table customer

# now tell hive where to find the informix data

# to get to the hive command prompt just type in hive
```

```
$ hive
Hive history file=/tmp/cloudera/yada_yada_log123.txt
hive>

# here is the hiveql you need to create the tables
# using a file is easier than typing

create external table customer (
cn int,
fname string,
lname string,
company string,
addr1 string,
addr2 string,
city string,
state string,
zip string,
phone string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/cloudera/customer'
;

# we already imported the db2 staff table above

# now tell hive where to find the db2 data
create external table staff (
id int,
name string,
dept string,
job string,
years string,
salary float,
comm float)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/cloudera/staff'
;

# you can put the commands in a file
# and execute them as follows:

$ hive -f hivestaff
Hive history file=/tmp/cloudera/hive_job_log_cloudera_201208101502_2140728119.txt
OK
Time taken: 3.247 seconds
OK
10 Sanders 20 Mgr   7 98357.5 NULL
20 Pernal 20 Sales 8 78171.25 612.45
30 Marenghi 38 Mgr   5 77506.75 NULL
40 O'Brien 38 Sales 6 78006.0 846.55
50 Hanes 15 Mgr   10 80
... lines deleted

# now for the join we've all been waiting for :-)

# this is a simple case, Hadoop can scale well into the petabyte range!

$ hive
Hive history file=/tmp/cloudera/hive_job_log_cloudera_201208101548_497937669.txt
hive> select customer.cn, staff.name,
> customer.addr1, customer.city, customer.phone
> from staff join customer
> on ( staff.id = customer.cn );
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
set hive.exec.reducers.bytes.per.reducer=number
```

```
In order to limit the maximum number of reducers:
set hive.exec.reducers.max=number
In order to set a constant number of reducers:
set mapred.reduce.tasks=number
Starting Job = job_201208101425_0005,
Tracking URL = http://0.0.0.0:50030/jobdetails.jsp?jobid=job_201208101425_0005
Kill Command = /usr/lib/hadoop/bin/hadoop
job  -Dmapred.job.tracker=0.0.0.0:8021 -kill job_201208101425_0005
2012-08-10 15:49:07,538 Stage-1 map = 0%,  reduce = 0%
2012-08-10 15:49:11,569 Stage-1 map = 50%,  reduce = 0%
2012-08-10 15:49:12,574 Stage-1 map = 100%,  reduce = 0%
2012-08-10 15:49:19,686 Stage-1 map = 100%,  reduce = 33%
2012-08-10 15:49:20,692 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201208101425_0005
OK
110 Ngan 520 Topaz Way        Redwood City    415-743-3611
120 Naughton 6627 N. 17th Way     Phoenix         602-265-8754
Time taken: 22.764 seconds
```

Es mucho más bonito cuando usa Hue para una interfaz de navegador gráfica, como se muestra en las Figuras 9, 10, y 11.

Figura 9. Hue Beeswax GUI para Hive en CDH4, ver consulta Hiveql

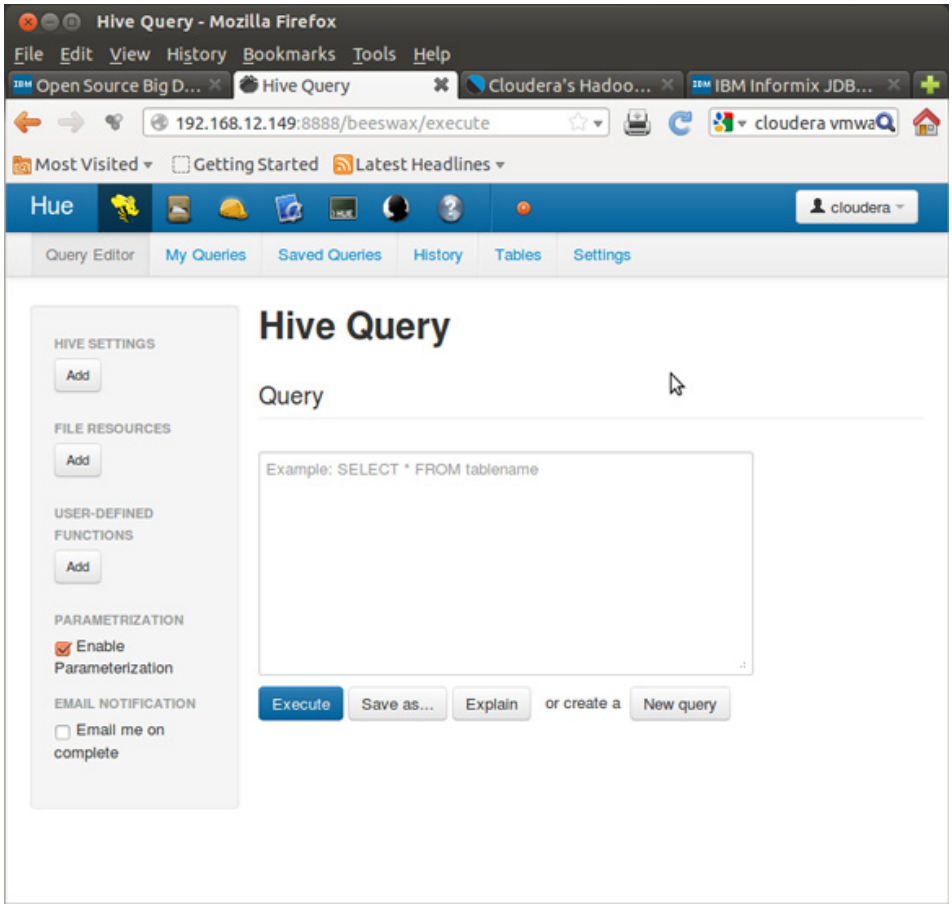


Figura 10. Hue Beeswax GUI para Hive, ver consulta Hiveql

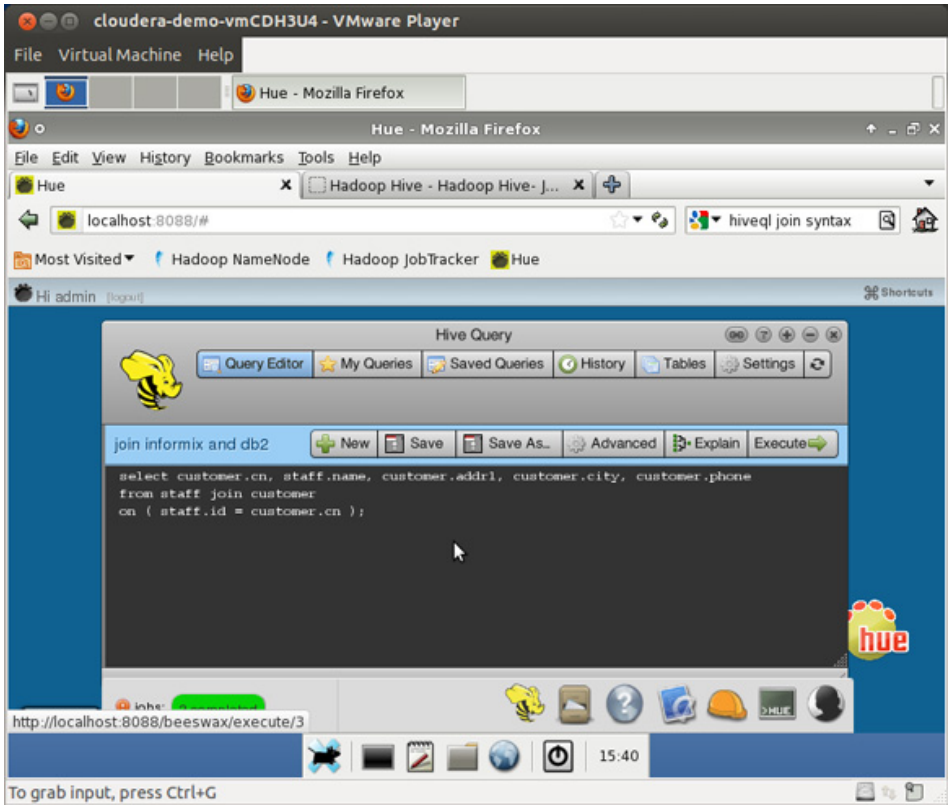
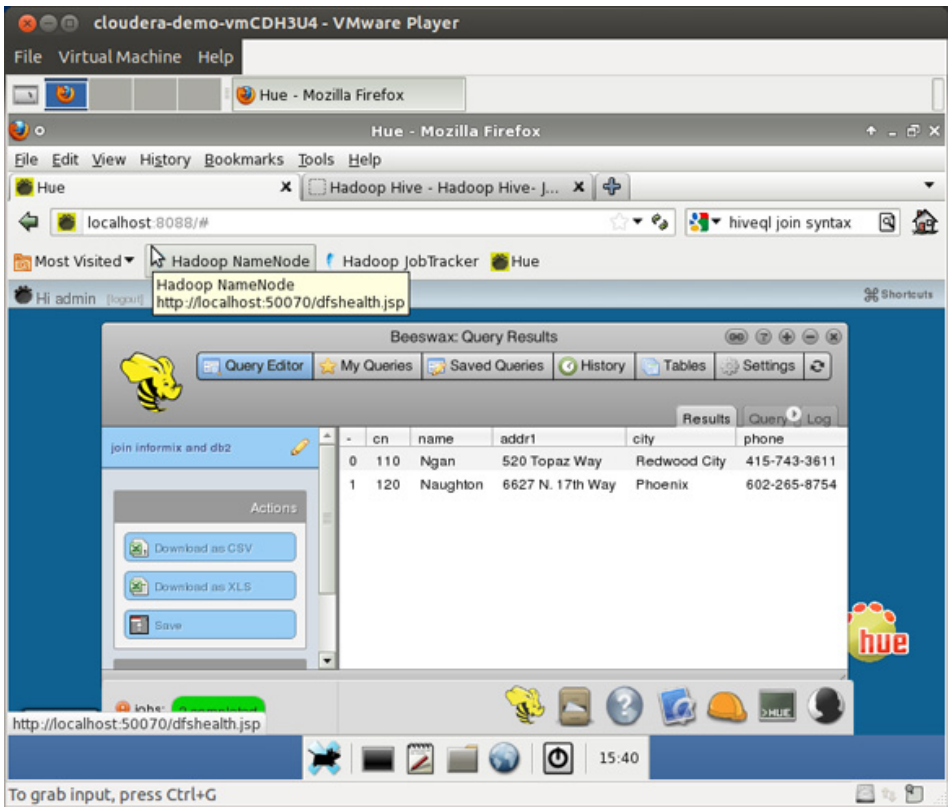


Figura 11. Interfaz gráfica Hue Beeswax, ver resultado de unión Informix-DB2



Usando Pig: Uniendo datos de Informix y DB2

Pig es un lenguaje de procedimiento. Justo como Hive, bajo las cubiertas que genera código MapReduce. La facilidad de uso de Hadoop continuará mejorando conforme más proyectos están disponibles. De la misma forma que a nosotros realmente nos gusta la línea de comando, existen varias intefaces gráficas de usuarios que funcionan muy bien con Hadoop.

El Listado 19 muestra el código Pig que es usado para unir la tabla del cliente y la tabla del personal del ejemplo previo.

Listado 19. Ejemplo Pig para unir la tabla Informix a la tabla DB2

```
$ pig
grunt> staffdb2 = load 'staff' using PigStorage(',')
>> as ( id, name, dept, job, years, salary, comm );
grunt> custifx2 = load 'customer' using PigStorage(',') as
>>  (cn, fname, lname, company, addr1, addr2, city, state, zip, phone)
>> ;
grunt> joined = join custifx2 by cn,  staffdb2 by id;

# to make pig generate a result set use the dump command
# no work has happened up till now

grunt> dump joined;
2012-08-11 21:24:51,848 [main] INFO  org.apache.pig.tools.pigstats.ScriptState
- Pig features used in the script: HASH_JOIN
2012-08-11 21:24:51,848 [main] INFO  org.apache.pig.backend.hadoop.executionengine
.HExecutionEngine - pig.usenewlogicalplan is set to true.
New logical plan will be used.

HadoopVersion PigVersion UserId StartedAt FinishedAt Features
0.20.2-cdh3u4 0.8.1-cdh3u4 cloudera 2012-08-11 21:24:51
2012-08-11 21:25:19 HASH_JOIN

Success!

Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTIme AvgMapTime
MaxReduceTime MinReduceTime AvgReduceTime Alias Feature Outputs
job_201208111415_0006 2 1 8 8 8 10 10 10
custifx,joined,staffdb2 HASH_JOIN hdfs://0.0.0.0/tmp/temp1785920264/tmp-388629360,

Input(s):
Successfully read 35 records from: "hdfs://0.0.0.0/user/cloudera/staff"
Successfully read 28 records from: "hdfs://0.0.0.0/user/cloudera/customer"

Output(s):
Successfully stored 2 records (377 bytes) in:
"hdfs://0.0.0.0/tmp/temp1785920264/tmp-388629360"

Counters:
Total records written : 2
Total bytes written : 377
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_201208111415_0006

2012-08-11 21:25:19,145 [main] INFO
org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2012-08-11 21:25:19,149 [main] INFO  org.apache.hadoop.mapreduce.lib.
```

```
input.FileInputFormat - Total input paths to process : 1
2012-08-11 21:25:19,149 [main] INFO  org.apache.pig.backend.hadoop.
executionengine.util.MapRedUtil - Total input paths to process : 1
(110,Roy            ,Jaeger         ,AA Athletics        ,520 Topaz Way
,null,Redwood City   ,CA,94062,415-743-3611      ,110,Ngan,15,Clerk,5,42508.20,206.60)
(120,Fred           ,Jewell         ,Century Pro Shop    ,6627 N. 17th Way
,null,Phoenix        ,AZ,85016,602-265-8754
,120,Naughton,38,Clerk,null,42954.75,180.00)
```

¿Cómo escojo Java, Hive, o Pig?

Usted tiene múltiples opciones para programar Hadoop, y es mejor ver el caso de uso para seleccionar la herramienta adecuada para el trabajo. Usted no está limitado a trabajar con datos relacionales pero este artículo está enfocado en Informix, DB2, y Hadoop funcionan do en conjunto bien. El escribir cientos de líneas en Java para implementar una unión hash estilo relacional es una completa pérdida de tiempo ya que este algoritmo de Hadoop MapReduce ya está disponible. ¿Cómo escoge? Esta es una cuestión de preferencia personal. A algunos les gustan las operaciones de configuración de códigos en SQL. Otros prefieren el código de procedimientos. Usted debe escoger el lenguaje que lo haga más productivo. Si usted tiene múltiples sistemas relacionales y desea combinar todos los datos con un gran rendimiento a un bajo punto de precio, Hadoop, MapReduce, Hive, y Pig están listos para ayudar.

No suprima sus datos: Recorriendo una partición de Informix hacia HDFS

Las bases de datos relacionales más modernas pueden particionar datos. Un caso de uso común es particionar por período de tiempo. Una ventana fija de datos es almacenada, por ejemplo un intervalo de 18 meses que se recorre, después del que los datos son archivados. La posibilidad de separar-particionar es muy poderosa. ¿Pero después de que la partición es separada qué hace uno con los datos?

El colocar en cinta los archivos de los datos viejos es una manera muy cara de descartar los bytes viejos. Una vez movidos a un medio menos aceptable, los datos son muy raramente accedidos a menos que haya un requerimiento de auditoría legal. Hadoop proporciona una alternativa mucho mejor.

El mover los bytes de archivado de la vieja partición hacia el acceso de alto rendimiento de Hadoop proporciona acceso de alto desempeño con un costo mucho menor que mantener los datos en el sistema original de transacciones o datamart/depósito de datos. Los datos don demasiado viejos para ser de valor de transacción, pero todavía son muy valiosos para la organización para análisis de largo plazo. Los ejemplos de Sqoop mostrados previamente proporcionan los puntos básicos para mover estos datos desde una partición relacional hacia HDFS.

Fuse - Obteniendo sus archivos HDFS vía NFS

Los datos de Informix/DB2/archivo plano en HDFS pueden accederse vía NFS, como se muestra en el Listado 20. Esto proporciona operaciones de línea de comando usando la interfaz "hadoop fs -yadayada". Desde una perspectiva tecnológica de caso de uso, NFS tiene limitaciones severas en el entorno de Big Data, pero los ejemplos están incluidos para desarrolladores y datos no tan grandes.

Listado 20. Configurando Fuse - acceda su datos HDFS vía NFS

```
# this is for CDH4, the CDH3 image doesn't have fuse installed...
$ mkdir fusemnt
$ sudo hadoop-fuse-dfs dfs://localhost:8020 fusemnt/
INFO fuse_options.c:162 Adding FUSE arg fusemnt/
$ ls fusemnt
tmp  user  var
$ ls fusemnt/user
cloudera  hive
$ ls fusemnt/user/cloudera
customer  DS.txt.gz  HF.out  HF.txt  orders  staff
$ cat fusemnt/user/cloudera/orders/part-m-00001
1007,2008-05-31,117,null,n,278693    ,2008-06-05,125.90,25.20,null
1008,2008-06-07,110,closed Monday
,y,LZ230     ,2008-07-06,45.60,13.80,2008-07-21
1009,2008-06-14,111,next door to grocery
,n,4745      ,2008-06-21,20.40,10.00,2008-08-21
1010,2008-06-17,115,deliver 776 King St. if no answer
,n,429Q      ,2008-06-29,40.60,12.30,2008-08-22
1011,2008-06-18,104,express
,n,B77897    ,2008-07-03,10.40,5.00,2008-08-29
1012,2008-06-18,117,null,n,278701    ,2008-06-29,70.80,14.20,null
```

Flume - cree un archivo listo para carga

Próxima generación de Flume, o flume-ng es un cargador paralelo de alta velocidad. Las bases de datos tienen cargadores de alta velocidad, ¿así que cómo trabajan bien conjuntamente? El caso de uso relacional para Flume-ng está creando un archivo listo para carga, localmente o remotamente, así que un servidor relacional puede usar este cargador de alta velocidad. Sí,esta funcionalidad se sobrepone a Sqoop, pero el script mostrado en el Listado 21 fue creado a solicitud de un cliente específicamente para este estilo de carga de base de datos.

Listado 21. Exportando datos HDFS hacia un archivo plano para cargar por cualquier base de datos

```
$  sudo yum install flume-ng

$ cat flumeconf/hdfs2dbloadfile.conf
#
# started with example from flume-ng documentation
# modified to do hdfs source to file sink
#

# Define a memory channel called ch1 on agent1
 agent1.channels.ch1.type = memory

# Define an exec source called exec-source1 on agent1 and tell it
# to bind to 0.0.0.0:31313. Connect it to channel ch1.
agent1.sources.exec-source1.channels = ch1
agent1.sources.exec-source1.type = exec
agent1.sources.exec-source1.command =hadoop fs -cat /user/cloudera/orders/part-m-00001
# this also works for all the files in the hdfs directory
# agent1.sources.exec-source1.command =hadoop fs
# -cat /user/cloudera/tsortin/*
agent1.sources.exec-source1.bind = 0.0.0.0
agent1.sources.exec-source1.port = 31313

# Define a logger sink that simply file rolls
# and connect it to the other end of the same channel.
agent1.sinks.fileroll-sink1.channel = ch1
agent1.sinks.fileroll-sink1.type = FILE_ROLL
agent1.sinks.fileroll-sink1.sink.directory =/tmp
```

```
# Finally, now that we've defined all of our components, tell
# agent1 which ones we want to activate.
agent1.channels = ch1
agent1.sources = exec-source1
agent1.sinks = fileroll-sink1

# now time to run the script

$ flume-ng agent --conf ./flumeconf/ -f ./flumeconf/hdfs2dbloadfile.conf -n
agent1

# here is the output file
# don't forget to stop flume - it will keep polling by default and generate
# more files

$ cat /tmp/1344780561160-1
1007,2008-05-31,117,null,n,278693    ,2008-06-05,125.90,25.20,null
1008,2008-06-07,110,closed Monday ,y,LZ230     ,2008-07-06,45.60,13.80,2008-07-21
1009,2008-06-14,111,next door to  ,n,4745      ,2008-06-21,20.40,10.00,2008-08-21
1010,2008-06-17,115,deliver 776 King St. if no answer       ,n,429Q
,2008-06-29,40.60,12.30,2008-08-22
1011,2008-06-18,104,express     ,n,B77897    ,2008-07-03,10.40,5.00,2008-08-29
1012,2008-06-18,117,null,n,278701    ,2008-06-29,70.80,14.20,null

# jump over to dbaccess and use the greatest
# data loader in informix: the external table
# external tables were actually developed for
# informix XPS back in the 1996 timeframe
# and are now available in may servers

#
drop table eorders;
create external table eorders
(on char(10),
mydate char(18),
foo char(18),
bar char(18),
f4 char(18),
f5 char(18),
f6 char(18),
f7 char(18),
f8 char(18),
f9 char(18)
)
using (datafiles ("disk:/tmp/myfoo" ) , delimiter ",");
select * from eorders;
```

Oozie - adding work flow for multiple jobs

Oozie will chain together multiple Hadoop jobs. There is a nice set of examples included with oozie that are used in the code set shown in Listing 22.

Listado 22. Job control with oozie

```
# This sample is for CDH3

# untar the examples

# CDH4
$ tar -zxvf /usr/share/doc/oozie-3.1.3+154/oozie-examples.tar.gz

# CDH3
$ tar -zxvf /usr/share/doc/oozie-2.3.2+27.19/oozie-examples.tar.gz

# cd to the directory where the examples live
```

```
# you MUST put these jobs into the hdfs store to run them

$  hadoop fs -put examples examples

# start up the oozie server - you need to be the oozie user
# since the oozie user is a non-login id use the following su trick

# CDH4
$ sudo su - oozie -s /usr/lib/oozie/bin/oozie-sys.sh start

# CDH3
$ sudo su - oozie -s /usr/lib/oozie/bin/oozie-start.sh

# checkthe status
oozie admin -oozie http://localhost:11000/oozie -status
System mode: NORMAL

# some jar housekeeping so oozie can find what it needs

$ cp /usr/lib/sqoop/sqoop-1.3.0-cdh3u4.jar examples/apps/sqoop/lib/
$ cp /home/cloudera/Informix_JDBC_Driver/lib/ifxjdbc.jar examples/apps/sqoop/lib/
$ cp /home/cloudera/Informix_JDBC_Driver/lib/ifxjdbcx.jar examples/apps/sqoop/lib/

# edit the workflow.xml  file to use your relational database:

##################################
<command> import
--driver com.informix.jdbc.IfxDriver
--connect jdbc:informix-sqli://192.168.1.143:54321/stores_demo:informixserver=ifx117
--table orders --username informix --password useyours
--target-dir /user/${wf:user()}/${examplesRoot}/output-data/sqoop --verbose<command>
##################################

# from the directory where you un-tarred the examples file do the following:

$ hrmr examples;hput examples examples

# now you can run your sqoop job by submitting it to oozie

$  oozie job -oozie http://localhost:11000/oozie -config  \
    examples/apps/sqoop/job.properties -run

job: 0000000-120812115858174-oozie-oozi-W

# get the job status from the oozie server

$ oozie job -oozie http://localhost:11000/oozie -info 0000000-120812115858174-oozie-oozi-W
Job ID : 0000000-120812115858174-oozie-oozi-W
------------------------------------------------------------------------
Workflow Name : sqoop-wf
App Path      : hdfs://localhost:8020/user/cloudera/examples/apps/sqoop/workflow.xml
Status        : SUCCEEDED
Run           : 0
User          : cloudera
Group         : users
Created       : 2012-08-12 16:05
Started       : 2012-08-12 16:05
Last Modified : 2012-08-12 16:05
Ended         : 2012-08-12 16:05

Actions
------------------------------------------------------------------------
ID       Status    Ext ID                 Ext Status Err Code
------------------------------------------------------------------------
0000000-120812115858174-oozie-oozi-W@sqoop-node                              OK
job_201208120930_0005  SUCCEEDED  -
------------------------------------------------------------------------
```

```
# how to kill a job may come in useful at some point

oozie job -oozie http://localhost:11000/oozie -kill
0000013-120812115858174-oozie-oozi-W

# job output will be in the file tree
$ hcat /user/cloudera/examples/output-data/sqoop/part-m-00003
1018,2008-07-10,121,SW corner of Biltmore Mall              ,n,S22942
,2008-07-13,70.50,20.00,2008-08-06
1019,2008-07-11,122,closed till noon Mondays                 ,n,Z55709
,2008-07-16,90.00,23.00,2008-08-06
1020,2008-07-11,123,express                                  ,n,W2286
,2008-07-16,14.00,8.50,2008-09-20
1021,2008-07-23,124,ask for Elaine                           ,n,C3288
,2008-07-25,40.00,12.00,2008-08-22
1022,2008-07-24,126,express                                  ,n,W9925
,2008-07-30,15.00,13.00,2008-09-02
1023,2008-07-24,127,no deliveries after 3 p.m.               ,n,KF2961
,2008-07-30,60.00,18.00,2008-08-22


# if you run into this error there is a good chance that your
# database lock file is owned by root
$  oozie job -oozie http://localhost:11000/oozie -config \
examples/apps/sqoop/job.properties -run

Error: E0607 : E0607: Other error in operation [<openjpa-1.2.1-r752877:753278
fatal store error> org.apache.openjpa.persistence.RollbackException:
The transaction has been rolled back.  See the nested exceptions for
details on the errors that occurred.], {1}

# fix this as follows
$ sudo chown oozie:oozie  /var/lib/oozie/oozie-db/db.lck

# and restart the oozie server
$ sudo su - oozie -s /usr/lib/oozie/bin/oozie-stop.sh
$ sudo su - oozie -s /usr/lib/oozie/bin/oozie-start.sh
```

HBase, es un almacenamiento de valor clave de alto rendimiento

HBase, es un almacenamiento de valor clave de alto rendimiento Si su caso de uso requiere escalabilidad y solo requiere el equivalente de base de datos de transacciones auto-confirmación, HBase puede bien ser la tecnología a emplear. HBase no es una base de datos. El nombre es desafortunado ya que para algunos, el término base implica base de datos. Pero realiza un excelente trabajo para almacenamientos de valor clave de alto rendimiento. Existe alguna sobreposición entre la funcionalidad de HBase, Informix, DB2 y otras bases de datos relacionales. Para transacciones ACID, conformidad SQL completa, y múltiples índices una base de datos tradicional es la opción obvia.

Este último ejercicio de código tiene la intención de dar familiaridad básica con HBASE. Es sencillo por diseño y de ninguna manera representa el alcance de la funcionalidad de HBase. Por favor use este ejemplo para comprender algunas de las posibilidades básicas en HBase. "HBase, The Definitive Guide", por Lars George, es obligatorio de leer si planea implementar o rechazar HBase para su caso de uso particular.

Este último ejemplo, mostrado en los Listados 23 y 24, usa la interfaz REST proporcionada con HBase para insertar valores claves en una tabla HBase. El armazón de prueba está basado en espiral.

Listado 23. Cree una tabla HBase e inserte una fila

```
# enter the command line shell for hbase

$ hbase shell
HBase Shell; enter 'help<RETURN> for list of supported commands.
Type "exit<RETURN> to leave the HBase Shell
Version 0.90.6-cdh3u4, r, Mon May  7 13:14:00 PDT 2012

#  create a table with a single column family

hbase(main):001:0> create 'mytable', 'mycolfamily'

# if you get errors from hbase you need to fix the
# network config

# here is a sample of the error:

ERROR: org.apache.hadoop.hbase.ZooKeeperConnectionException: HBase
is able to connect to ZooKeeper but the connection closes immediately.
This could be a sign that the server has too many connections
(30 is the default). Consider inspecting your ZK server logs for
that error and then make sure you are reusing HBaseConfiguration
as often as you can. See HTable's javadoc for more information.

# fix networking:

# add the eth0 interface to /etc/hosts with a hostname

$ sudo su -
# ifconfig | grep addr
eth0      Link encap:Ethernet  HWaddr 00:0C:29:8C:C7:70
inet addr:192.168.1.134  Bcast:192.168.1.255  Mask:255.255.255.0
Interrupt:177 Base address:0x1400
inet addr:127.0.0.1  Mask:255.0.0.0
[root@myhost ~]# hostname myhost
[root@myhost ~]# echo "192.168.1.134 myhost" >gt; /etc/hosts
[root@myhost ~]# cd /etc/init.d

# now that the host and address are defined restart Hadoop

[root@myhost init.d]# for i in hadoop*
> do
> ./$i restart
> done

# now try table create again:

$ hbase shell
HBase Shell; enter 'help<RETURN> for list of supported commands.
Type "exit<RETURN> to leave the HBase Shell
Version 0.90.6-cdh3u4, r, Mon May  7 13:14:00 PDT 2012

hbase(main):001:0> create 'mytable' , 'mycolfamily'
0 row(s) in 1.0920 seconds

hbase(main):002:0>

# insert a row into the table you created
# use some simple telephone call log data
# Notice that mycolfamily can have multiple cells
```

```
# this is very troubling for DBAs at first, but
# you do get used to it

hbase(main):001:0>  put 'mytable',  'key123', 'mycolfamily:number','6175551212'
0 row(s) in 0.5180 seconds
hbase(main):002:0>  put 'mytable',  'key123', 'mycolfamily:duration','25'

# now describe and then scan the table

hbase(main):005:0> describe 'mytable'
DESCRIPTION                                          ENABLED
{NAME => 'mytable', FAMILIES => [{NAME => 'mycolfam true
ily', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '
0', COMPRESSION => 'NONE', VERSIONS => '3', TTL =>
'2147483647', BLOCKSIZE => '65536', IN_MEMORY => 'f
alse', BLOCKCACHE => 'true'}]}
1 row(s) in 0.2250 seconds

#  notice that timestamps are included

hbase(main):007:0> scan 'mytable'
ROW                      COLUMN+CELL
key123                   column=mycolfamily:duration,
timestamp=1346868499125, value=25
key123                   column=mycolfamily:number,
timestamp=1346868540850, value=6175551212
1 row(s) in 0.0250 seconds
```

Listado 24. Usando la interfaz HBase REST

```
# HBase includes a REST server

$ hbase rest start -p 9393 &

# you get a bunch of messages....

# get the status of the HBase server

$ curl http://localhost:9393/status/cluster

# lots of output...
# many lines deleted...

mytable,,1346866763530.a00f443084f21c0eea4a075bbfdfc292.
stores=1
storefiless=0
storefileSizeMB=0
memstoreSizeMB=0
storefileIndexSizeMB=0

# now scan the contents of mytable

$ curl http://localhost:9393/mytable/*

# lines deleted
12/09/05 15:08:49 DEBUG client.HTable$ClientScanner:
Finished with scanning at REGION =>
# lines deleted
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CellSet><Row key="a2V5MTIz">
<Cell timestamp="1346868499125" column="bXljb2xmYW1pbHk6ZHVyYXRpb24=">MjU=</Cell>
<Cell timestamp="1346868540850" column="bXljb2xmYW1pbHk6bnVtYmVy">NjE3NTU1MTIxMg==</Cell>
<Cell timestamp="1346868425844" column="bXljb2xmYW1pbHk6bnVtYmVy">NjE3NTU1MTIxMg==</Cell>
</Row></CellSet>

# the values from the REST interface are base64 encoded
```

```
$ echo a2V5MTIz | base64 -d
key123
$ echo bXljb2xmYW1pbHk6bnVtYmVy | base64 -d
mycolfamily:number

# The table scan above gives the schema needed to insert into the HBase table

$ echo RESTinsertedKey | base64
UkVTVGluc2VydGVkS2V5Cg==

$ echo 7815551212 | base64
NzgxNTU1MTIxMgo=

# add a table entry with a key value of "RESTinsertedKey" and
# a phone number of "7815551212"

# note - curl is all on one line
$  curl -H "Content-Type: text/xml" -d '<CellSet>
<Row key="UkVTVGluc2VydGVkS2V5Cg==">
<Cell column="bXljb2xmYW1pbHk6bnVtYmVy">NzgxNTU1MTIxMgo=<Cell>
<Row><CellSet> http://192.168.1.134:9393/mytable/dummykey

12/09/05 15:52:34 DEBUG rest.RowResource: POST http://192.168.1.134:9393/mytable/dummykey
12/09/05 15:52:34 DEBUG rest.RowResource: PUT row=RESTinsertedKey\x0A,
families={(family=mycolfamily,
keyvalues=(RESTinsertedKey\x0A/mycolfamily:number/9223372036854775807/Put/vlen=11)}

# trust, but verify

hbase(main):002:0> scan 'mytable'
ROW                  COLUMN+CELL
RESTinsertedKey\x0A column=mycolfamily:number,timestamp=1346874754883,value=7815551212\x0A
key123               column=mycolfamily:duration, timestamp=1346868499125, value=25
key123               column=mycolfamily:number, timestamp=1346868540850, value=6175551212
2 row(s) in 0.5610 seconds

# notice the \x0A at the end of the key and value
# this is the newline generated by the "echo" command
# lets fix that

$ printf 8885551212 | base64
ODg4NTU1MTIxMg==

$ printf mykey | base64
bXlrZXk=

# note - curl statement is all on one line!
curl -H "Content-Type: text/xml" -d '<CellSet><Row key="bXlrZXk=">
<Cell column="bXljb2xmYW1pbHk6bnVtYmVy">ODg4NTU1MTIxMg==<Cell>
<Row><CellSet>
http://192.168.1.134:9393/mytable/dummykey

# trust but verify
hbase(main):001:0> scan 'mytable'
ROW                    COLUMN+CELL
RESTinsertedKey\x0A column=mycolfamily:number,timestamp=1346875811168,value=7815551212\x0A
key123               column=mycolfamily:duration, timestamp=1346868499125, value=25
key123               column=mycolfamily:number, timestamp=1346868540850, value=6175551212
mykey                column=mycolfamily:number, timestamp=1346877875638, value=8885551212
3 row(s) in 0.6100 seconds
```

Conclusión

Ohh, usted llegó hasta el final, ¡felicidades! Esto es solo el inicio de la comprensión de Hadoop y de cómo interactua con Informix y DB2. Aquí hay algunas sugerencias para sus próximos pasos.

- Tome los ejemplos mostrados previamente y adáptelos a sus servidores. Usted deseará usar pocos datos ya que no hay mucho espacio en la imagen virtual.
- Certifíquese como un Administrador Hadoop. Visite el sitio Cloudera para saber de cursos e información de prueba.
- Certifíquese como un Desarrollador Hadoop.
- Inicie un clúster usando la edición gratuita de Cloudera Manager.
- Inicie con IBM Big Sheets ejecutándose sobre CDH4.

Recursos

Aprender

- Utilice un RSS feed para solicitar una notificación de los próximos artículos en esta serie. (Conozca más sobre RSS feeds of developerWorks content.)
- Aprenda más acerca de Hadoop, Open Source, e Indemnification con demos gratuitos de Cloudera Hadoop.
- Descargue demos de Cloudera Hadoop.
- Sepa más acerca de la Sociedad de IBM y Cloudera al leer el artículo de developerWorks"Integración a Nivel de Hadoop: BigInsights de IBM ahora da soporte a Cloudera" .
- Lea el libro "Hadoop the Definitive Guide" por Tom White, 3a edición.
- Lea el libro "Programming Hive" por Edward Capriolo, Dean Wampler, y Jason Rutherglen.
- Lea el libro "HBase the Definitive Guide" por Lars George.
- Lea el libro "Programming Pig" por Alan Gates.
- Lea otros artículos de developerWorks por Marty Lurie.
- Visite el Wiki zona developerWorks Information Management para encontrar más recursos para desarrolladores y administradores de DB2.
- Manténgase al tanto de los Eventos técnicos y webcasts de developerWorks enfocados en una variedad de productos IBM y de temas de la industria TI.
- Asista a una gratuita de gratuita para actualizarse rápidamente sobre productos y herramientas IBM y sobre las tendencias de la industria de la TI.
- Siga a developerWorks en Twitter.
- Vigile las demos on demand de developerWorks que van desde demostraciones sobre instalación y configuración de productos para principiantes, hasta funcionalidades avanzadas pera desarrolladores experimentados.

Obtener los productos y tecnologías

- Obtenga descargas gratuitas de Cloudera Hadoop.
- Descargue Informix Developer edition.
- Descargue DB2 Express-C edition.
- Instale Informix.
- Instale el controlador DB2 desde IBM Software .
- Instale el controlador DB2 desde IBM Support Portal.
- Construya su próximo proyecto de desarrollo con software de prueba IBM, disponible para descarga directamente de developerWorks.
- Evalúe los productos de IBM como mejor le parezca: descargue una prueba de producto, pruebe un producto online, use un producto en un entorno de nube o invierta unas cuantas horas en el Recinto de Seguridad de la SOA aprendiendo a implementar la Arquitectura Orientada a Servicios con eficiencia.

Comentar

- Participe en My developerWorks. Conéctese con otros usuarios de developerWorks mientras explora los blogs conducidos por desarrolladores, foros, grupos y wikis.

Sobre el autor

Marty Lurie

Marty Lurie inició su carrera generando tarjetas de perforación mientras se intentaba escribir Fortran en IBM 1130. Su trabajo cotidiano consiste en Ingeniería de Sistemas Hadoop en Cloudera, pero si se le presiona admite que básicamente juega con computadoras. Su programa favorito es el que escribió para conectar su Nordic Track a su laptop (la laptop perdió dos libras, y bajó su colesterol en un 20%). Marty está certificado como Administrador Hadoop Certificado por Cloudera, Desarrollador Hadoop Certificado por Cloudera, Administrador WebSphere Avanzado certificado por IBM, Profesional certificado por Informix, Certificado en DB2 DBA, Profesional Certificado en Soluciones de Inteligencia de Negocios, Certificado en Linux+, y ha entrenado a su perro a jugar a baloncesto. Usted puede ponerse en contacto con Marty en marty@cloudera.com.