**Báo cáo cơ sở dữ liệu**

**Hệ quản trị CSDL HIVE**

***Nhóm 14***

Thành viên:

Nguyễn Thị Phương (14020358)

Nguyễn Thanh Long (14020261)

Bùi Thị Lan Hương (14020227)

Phụ Lục

[**I.** **Hadoop** 1](#_Toc468390806)

[**II.** **HIVE là gì?** 2](#_Toc468390809)

[**III.** **Cài đặt.** 4](#_Toc468390810)

[**1.** **Yêu cầu** 4](#_Toc468390811)

[**2.** **Bước thực hiện** 4](#_Toc468390812)

**IV. Thao tác với HIVE**

**Giới thiệu**

'Big Data' được sử dụng cho việc thu thập các tập dữ liệu lớn bao gồm khối lượng lớn, tốc độ cao, và một loạt các dữ liệu đang tăng lên từng ngày. Sử dụng hệ thống quản lý dữ liệu truyền thống, rất khó để xử lý 'Big Data'. Do đó, Apache Software Foundation đã giới thiệu một khuôn khổ gọi là Hadoop để giải quyết vấn quản lý 'Big Data' và những thử thách của tiến trình.

1. **Hadoop**

Apache Hadoop là 1 nền tảng mềm mã nguồn mở, miễn phí, dựa trên Java, cung cấp 1 nền tảng phân tán mạnh để lưu trữ và quản lý big data. Bản quyền Apache V2 license. Nó chạy ứng dụng trên các cụm phần cứng lớn và xử lý hàng ngàn terabytes dữ liệu trên hàng ngàn nodes. Hadoop lấy cảm hứng từ Google MapReduce và Google File System (GFS). Điểm mạnh của nền tảng Hadoop là cung cấp độ tin cậy và tính sẵn sàng cao.

## 1. Các thành phần lõi của Hadoop là gì?

Có hai thành phần chính của nền tảng Hadoop.

* Hadoop MapReduce là cách chia 1 vấn đề dữ liệu lớn hơn thành các đoạn nhỏ hơn và phân tán nó trên nhiều máy chủ. Mỗi máy chủ có 1 tập tài nguyên riêng và máy chủ xử lý dữ liệu trên cục bộ. Khi máy chủ xử lý xong dữ liệu, chúng sẽ gởi trở về máy chủ chính.
* Hadoop Distributed File System (HDFS) là hệ thống tập tin ảo. Có 1 sự khác biệt lớn giữa các hệ thống tập tin khác và Hadoop. Khi chúng ta di chuyển 1 tập tin trên HDFS, nó tự động chia thành nhiều mảnh nhỏ. Các đoạn nhỏ của tập tin sẽ được nhân rộng và lưu trữ trên nhiều máy chủ khác (thường là 3) để tăng sức chịu lỗi và tính sẵn sàng cao.

## 2. Các hệ sinh thái Hadoop chứa các tiểu dự án khác nhau (công cụ) như Sqoop, Pig, và Hive được sử dụng để hỗ trợ Hadoop.

+ Sqoop: Nó được sử dụng để nhập và xuất dữ liệu đến và đi giữa HDFS và RDBMS.

+ Pig: Đó là một nền tảng ngôn ngữ thủ tục được sử dụng để phát triển một kịch bản cho các hoạt động MapReduce.

+ Hive: Đó là một nền tảng được sử dụng để phát triển các kịch bản kiểu SQL để làm các hoạt động MapReduce.

1. **HIVE là gì?**
2. **Tổng quan về Hive**

Apache Hive là 1 kho dữ liệu (data warehouse) hỗ trợ người sử dụng có thể dễ dàng hơn trong việc quản lý và truy vấn đối với tập dữ liệu lớn được lưu trữ trên các hệ thống lưu trữ phân tán (distributed storage). Hive được xây dựng dựa trên cơ sở của Apache Hadoop, nó cung cấp các tính năng chính sau:

* Công cụ cho phép dễ dàng thực hiện tác vụ như trích xuất, vận chuyển và lưu trữ dữ liệu.
* Cơ chế để xử lý cho nhiều định dạng dữ liệu khác nhau.
* Truy cập tới dữ liệu dạng files được lưu trữ trực tiếp ở trong Apache HDFS hoặc đối với nhiều hệ thống lưu trữ dữ liệu khác như Apache Hbase.
* Thực hiện query thông qua MapReduce.

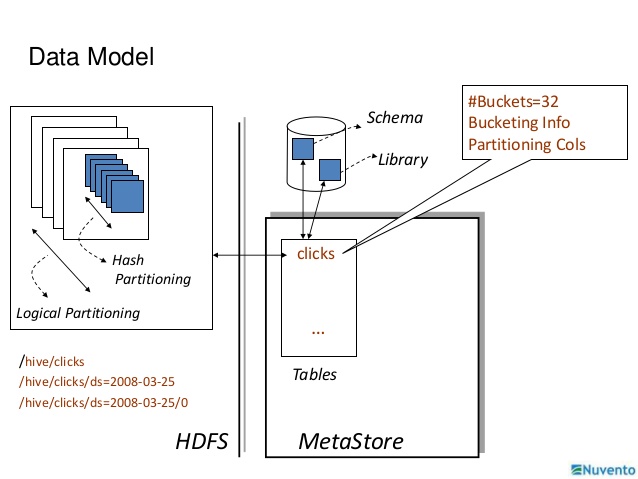
Hive định nghĩa ra 1 ngôn ngữ truy vấn đơn giản có cú pháp gần giống với SQL được gọi là HiveQL, nó cho phép người sử dụng đã quen thuộc với các truy vấn SQL thực hiện việc truy vấn dữ liệu. Ngoài ra ngôn ngữ này còn cho phép các lập trình viên người đã quen thuộc với MapReduce framework có thể nhúng các mappers và reducers cho chính họ viết ra để thực thi nhiều hơn nữa các phân tích phức tạp mà không được hỗ trợ bởi các hàm đã cho có sẵn trong ngôn ngữ HiveQL. HiveQL cũng có thể được mở rộng với các custom scalar functions, aggregations và các table functions.

Hive không yêu cầu dữ liệu phải được đọc ghi dưới một định dạng của riêng Hive (Hive format). Hive hoạt động tốt trên Thrift và các định dạng dữ liệu riêng của người sử dụng.

Hive không được thiết kế để cho các giao dịch online (OLTP workloads) và không nên dùng cho các real-time queries và các cập nhật trên từng dòng trong 1 table (row-level). Hive hoạt động tốt nhất cho các batch jobs trên các tập dữ liệu lớn, mà ở đó dữ liệu được thêm vào liên tục (append-only data), ví dụ như web logs. Hive có khả năng mở rộng theo chiều ngang tốt (thực thi tốt trên 1 hadoop cluster có số lượng máy biến đổi), có khả năng tích hợp với MapReduce framewok và UDF, UDAF, UDTF; có khả năng chống chịu lỗi và mềm dẻo đối với các dữ liệu đầu vào của chính nó.

Các thành phần cấu hình Hive bao gồm Hcatalog và WebHCat. Hcatalog là 1 thành phần của Hive. Đây là lớp quản lý lưu trữ cho Hadoop (table and management layer), nó cho phép người dùng với các công cụ xử lý dữ liệu khác nhau bao gồm cả Pig và MapReduce thực thi hoạt động đọc, ghi một cách dễ dàng hơn. WebHCat cung cấp 1 dịch vụ cho phép bạn có thể thực thi Hadoop MapReduce (hoặc Yarn), Pig, Hive.

Ban đầu Hive được phát triển bởi Facebook, sau đó Apache Software Foundation sử dụng nó và phát triển nó hơn nữa như một mã nguồn mở dưới tên Apache Hive. Nó được sử dụng bởi các công ty khác nhau. Ví dụ, Amazon sử dụng nó trong Amazon Elastic MapReduce.

1. **Mô hình dữ liệu:** 

Dữ liệu trong Hive được tổ chức thành các kiểu sau:

* Database: là namespace cho các tables, dùng để nhóm và quản lý các nhóm tables khác nhau.
* Tables: tương tự như table trong các hệ cơ sở dữ liệu quan hệ. Traong Hive table có thể thực hiện các phép toán filter, join và union... Mặc định thì dữ liệu của Hive sẽ được lưu bên trong thư mục warehouse trên HFDS. Tuy nhiên Hive cung cấp các kiểu external table cho phép ta tạo ra và quản lý các table mà dữ liệu của nó đã tồn tại từ trước khi tạo ra table này hoặc nó được lưu trữ ở 1 thư mục khác bên trong hệ thống HDFS. Tổ chức row và column bên trong Hive có nhiều điểm tương đồng với tổ chức row và column trong các hệ cơ sở dữ liệu quan hệ. Hive có 2 kiểu table đó là: Managed Table và External tables.
* Partitions: Mỗi table có thể có 1 hoặc nhiều các khóa mà từ đó xác định dữu liệu sẽ được lưu trữ ở đâu. Ví dụ table web\_log có thể phân chia dữ liệu của mình theo từng ngày là lưu dữ liệu của mỗi ngày trong 1 thư mục khác nhau bên dưới đường dẫn warehouse. Ví dụ: /warehouse/web\_log/date=”01-01-2014”.
* Buckets: Dữ liệu trong mỗi partition có thể phân chia thành nhiều buckets khác nhau dựa trên 1 hash của 1 column bên trong table. Mỗi bucket lưu trữ của nó bên dưới 1 thư mục riêng. Việc phân chia các partition thành các bucket giúp việc thực thi các query dễ dàng hơn.

1. **Cài đặt.**
   1. **Yêu cầu**

* Cài đặt trên môi trường Linux, hoặc Mac, Windows (windows sẽ khác và phức tạp hơn, nên chủ yếu là trên Linux)
* Máy có cài Java (JDK).
* Cài sẵn Hadoop
  1. **Bước thực hiện**

1. **Bước 1: Thẩm định Java.**

Java phải được cài đặt trên hệ thống của bạn trước khi cài đặt Hive. Hãy xác minh cài đặt java bằng cách sử dụng lệnh sau:

$ java –version

Nếu Java đã được cài đặt trên hệ thống của bạn, bạn sẽ thấy:

java version "1.7.0\_71"

Java(TM) SE RuntimeEnvironment(build 1.7.0\_71-b13)

JavaHotSpot(TM)Client VM (build 25.0-b02, mixed mode)

Nếu java không được cài đặt trong hệ thống của bạn, làm theo các bước dưới đây để cài đặt java.

## Cài đặt Java

### Bước I:

Tải java (JDK <phiên bản mới nhất> - X64.tar.gz) bằng cách truy cập vào liên kết sau đây [http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html.](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)

Sau đó jdk-7u71-linux-x64.tar.gz sẽ được tải vào hệ thống của bạn.

* Bước II:

Nói chung, bạn sẽ tìm thấy các tập tin java tải về trong thư mục Downloads. Xác minh và trích xuất các tập tin jdk-7u71-linux-x64.gz sử dụng các lệnh sau đây.

$ cd Downloads/

$ ls

jdk-7u71-linux-x64.gz

$ tar zxf jdk-7u71-linux-x64.gz

$ ls

jdk1.7.0\_71 jdk-7u71-linux-x64.gz

### Bước III:

Để làm cho java có sẵn cho tất cả người sử dụng, bạn phải di chuyển nó đến vị trí "/ usr / local /". Mở thư mục gốc, và gõ lệnh sau đây.

$ su

password:

# mv jdk1.7.0\_71 /usr/local/

# exit

### Bước IV:

Để thiết lập các biến PATH và JAVA\_HOME, thêm các lệnh sau đây trong ~ / .bashrc.

export JAVA\_HOME=/usr/local/jdk1.7.0\_71

export PATH=$PATH:$JAVA\_HOME/bin

Bây giờ áp dụng tất cả các thay đổi vào hệ thống đang chạy hiện nay.

$ source ~/.bashrc

### Bước V:

Sử dụng các lệnh sau để cấu hình lựa chọn thay thế java:

# alternatives --install /usr/bin/java java usr/local/java/bin/java 2

# alternatives --install /usr/bin/javac javac usr/local/java/bin/javac 2

# alternatives --install /usr/bin/jar jar usr/local/java/bin/jar 2

# alternatives --set java usr/local/java/bin/java

# alternatives --set javac usr/local/java/bin/javac

# alternatives --set jar usr/local/java/bin/jar

Bây giờ xác nhận cài đặt bằng cách sử dụng java -version lệnh từ các thiết bị đầu cuối như đã giải thích ở trên.

## 

## Bước 2: Cài đặt Hadoop

Hadoop phải được cài đặt trên hệ thống của bạn trước khi cài đặt Hive. Hãy xác minh cài đặt Hadoop sử dụng lệnh sau đây:

$ hadoop version

Nếu Hadoop đã được cài đặt trên hệ thống của bạn:

Hadoop2.4.1Subversion https://svn.apache.org/repos/asf/hadoop/common -r 1529768

Compiledby hortonmu on 2013-10-07T06:28Z

Compiledwith protoc 2.5.0

From source with checksum 79e53ce7994d1628b240f09af91e1af4

Nếu Hadoop không được cài đặt trên hệ thống của bạn, sau đó tiến hành các bước sau đây:

## Tải Hadoop

Tải về và giải nén Hadoop 2.4.1 từ Apache Software Foundation sử dụng các lệnh sau đây.

$ su

password:

# cd /usr/local

# wget http://apache.claz.org/hadoop/common/hadoop-2.4.1/

hadoop-2.4.1.tar.gz

# tar xzf hadoop-2.4.1.tar.gz

# mv hadoop-2.4.1/\* to hadoop/

# exit

## Cài đặt Hadoop trong chế độ Pseudo Distributed

Các bước sau đây được sử dụng để cài đặt Hadoop 2.4.1

### Bước I: Thiết lập Hadoop

Bạn có thể thiết lập các biến môi trường Hadoop bằng cách thêm các lệnh sau đây để **~ / .bashrc** file.

export HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/nativeexport

PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

Bây giờ áp dụng tất cả các thay đổi vào hệ thống đang chạy hiện nay.

$ source ~/.bashrc

### Bước II: Cấu hình Hadoop

Bạn có thể tìm thấy tất cả các tập tin cấu hình Hadoop ở vị trí "$ HADOOP\_HOME / etc / hadoop". Bạn cần phải thực hiện những thay đổi thích hợp trong các tập tin cấu hình theo sở hạ tầng Hadoop của bạn.

$ cd $HADOOP\_HOME/etc/hadoop

Để phát triển các chương trình Hadoop sử dụng java, bạn phải thiết lập lại các biến môi trường java trong tập tin **hadoop-env.sh**  bằng cách thay thế giá trị **JAVA\_HOME**  với vị trí của java trong hệ thống của bạn.

export JAVA\_HOME=/usr/local/jdk1.7.0\_71

- Dưới đây là danh sách các tập tin mà bạn phải chỉnh sửa để cấu hình Hadoop:

**core-site.xml**

tập tin **core-site.xml** chứa các thông tin như số cổng được sử dụng cho Hadoop dụ, cấp phát bộ nhớ cho hệ thống tập tin, giới hạn bộ nhớ để lưu trữ dữ liệu, và kích thước của read /write.

Mở core-site.xml và thêm các thuộc tính sau giữa <configuration> và </ configuration>.

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

**HDFS-site.xml**

Các **HDFS-site.xml** chứa thông tin như giá trị của dữ liệu sao chép, đường dẫn namenode, và đường dẫn datanode của hệ thống tập tin địa phương của bạn. Nó có nghĩa là nơi mà bạn muốn lưu trữ các sở hạ tầng Hadoop.

Mở tập tin này và thêm các thuộc tính sau giữa <configuration></ configuration> trong tập tin này.

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.name.dir</name>

<value>file:///home/hadoop/hadoopinfra/hdfs/namenode </value>

</property>

<property>

<name>dfs.data.dir</name>

<value>file:///home/hadoop/hadoopinfra/hdfs/datanode </value>

</property>

**Yarn-site.xml**

Mở tập tin yarn-site.xml và thêm các thuộc tính sau giữa <configuration></ configuration> trong tập tin này.

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

**Mapred-site.xml**

Tập tin này được sử dụng để xác định các MapReduce đang sử dụng. Theo mặc định, Hadoop có chứa một mẫu yarn-site.xml. Trước hết, bạn cần phải sao chép các tập tin từ mapred-site.xml.template đến tập tin mapred-site.xml bằng cách sử dụng lệnh sau đây.

$ cp mapred-site.xml.template mapred-site.xml

Mở **mapred-site.xml** và thêm các thuộc tính sau giữa <configuration></ configuration> trong tập tin này.

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

## 

## Xác minh cài đặt Hadoop

Các bước sau đây được sử dụng để xác nhận cài đặt Hadoop.

### Bước I: Cài đặt namenode

Thiết lập namenode cách sử dụng lệnh "HDFS namenode -format" như sau.

$ cd ~

$ hdfs namenode -format

Các kết quả dự kiến ​​như sau.

10/24/1421:30:55 INFO namenode.NameNode: STARTUP\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STARTUP\_MSG: Starting NameNode

STARTUP\_MSG: host = localhost/192.168.1.11

STARTUP\_MSG: args = [-format]

STARTUP\_MSG: version = 2.4.1

...

...

10/24/14 21:30:56 INFO common.Storage: Storage directory

/home/hadoop/hadoopinfra/hdfs/namenode has been successfully formatted.

10/24/14 21:30:56 INFO namenode.NNStorageRetentionManager: Going to

retain 1 images with txid >= 0

10/24/14 21:30:56 INFO util.ExitUtil: Exiting with status 0

10/24/14 21:30:56 INFO namenode.NameNode: SHUTDOWN\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SHUTDOWN\_MSG: Shutting down NameNode at localhost/192.168.1.11

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

### Bước II: Thẩm định dfs Hadoop

Các lệnh sau đây được sử dụng để bắt đầu dfs. Thực hiện lệnh này sẽ khởi động hệ thống tập tin Hadoop của bạn.

$ start-dfs.sh

Kết quả dự kiến ​​như sau:

10/24/1421:37:56

Starting namenodes on [localhost]

localhost: starting namenode, logging to /home/hadoop/hadoop-2.4.1/logs/hadoop-hadoop-namenode-localhost.out

localhost: starting datanode, logging to /home/hadoop/hadoop-2.4.1/logs/hadoop-hadoop-datanode-localhost.out

Starting secondary namenodes [0.0.0.0]

### Bước III: Thẩm định Yarn Script

Lệnh sau đây được sử dụng để bắt đầu yarn script

$ start-yarn.sh

Kết quả dự kiến ​​như sau:

starting yarn daemons

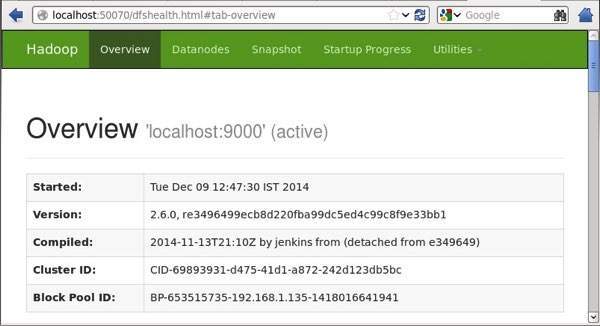
starting resourcemanager, logging to /home/hadoop/hadoop-2.4.1/logs/yarn-hadoop-resourcemanager-localhost.out

localhost: starting nodemanager, logging to /home/hadoop/hadoop-2.4.1/logs/yarn-hadoop-nodemanager-localhost.out

### Bước IV: Truy cập Hadoop trên trình duyệt

Số cổng mặc định để truy cập Hadoop là 50070. Sử dụng url sau đây để có được dịch vụ Hadoop trên trình duyệt của bạn.

http://localhost:50070/



### Bước V: Kiểm tra tất cả các ứng dụng cho cụm

Số cổng mặc định để truy cập tất cả các ứng dụng của cụm là 8088. Sử dụng url sau đây để ghé thăm dịch vụ này.

http://localhost:8088/



### 

### Bước 3: Tải Hive

Chúng tôi sử dụng hive-0.14.0 trong hướng dẫn này. Bạn có thể tải nó bằng cách truy cập vào liên kết sau đây [http://apache.petsads.us/hive/hive-0.14.0/.](http://apache.petsads.us/hive/hive-0.14.0/) Chúng ta giả sử nó được tải về vào thư mục / Downloads. Ở đây, chúng tôi tải về Hive lưu trữ có tên là "apache-hive-0.14.0-bin.tar.gz" cho hướng dẫn này. Các lệnh sau đây được sử dụng để xác minh download:

$ cd Downloads

$ ls

Thành công sẽ có:

apache-hive-0.14.0-bin.tar.gz

### 

### Bước 4: Cài đặt Hive

Các bước sau đây được yêu cầu để cài đặt Hive trên hệ thống của bạn.

### Trích xuất và xác minh Hive Lưu Trữ

Các lệnh sau đây được sử dụng để xác minh tải về và giải nén tệp tin hive:

$ tar zxvf apache-hive-0.14.0-bin.tar.gz

$ ls

Thành công sẽ xuất hiện trong list dưới:

apache-hive-0.14.0-bin apache-hive-0.14.0-bin.tar.gz

### Chuyển tập tin vào thư mục local / usr / hive

$ su -

passwd:

# cd /home/user/Download

# mv apache-hive-0.14.0-bin /usr/local/hive

# exit

### Thiết lập môi trường cho Hive

Bạn có thể thiết lập môi trường Hive bằng cách thêm các dòng sau vào **~ / .bashrc**:

export HIVE\_HOME=/usr/local/hive

export PATH=$PATH:$HIVE\_HOME/bin

export CLASSPATH=$CLASSPATH:/usr/local/Hadoop/lib/\*:.

export CLASSPATH=$CLASSPATH:/usr/local/hive/lib/\*:.

Sau đó:

$ source ~/.bashrc

### 

### Bước 5: Cấu hình Hive

Để cấu hình Hive với Hadoop, bạn cần phải chỉnh sửa **hive-env.sh** , được đặt trong **/ conf $ HIVE\_HOME**. Các lệnh sau chuyển hướng đến **hive/conf** và sao chép các tập tin template:

$ cd $HIVE\_HOME/conf

$ cp hive-env.sh.template hive-env.sh

Chỉnh sửa **hive-env.sh** tập tin bằng cách thêm các dòng sau:

export HADOOP\_HOME=/usr/local/hadoop

Cài đặt Hive kết thúc thành công. Bây giờ bạn cần một máy chủ cơ sở dữ liệu bên ngoài để cấu hình Metastore. Chúng tôi sử dụng cơ sở dữ liệu Apache Derby.

### Bước 6: Tải về và cài đặt Apache Derby

Thực hiện theo các bước dưới đây để tải về và cài đặt Apache Derby:

### Tải Apache Derby

Các lệnh sau đây được sử dụng để tải về Apache Derby.

$ cd ~

$ wget http://archive.apache.org/dist/db/derby/db-derby-10.4.2.0/db-derby-10.4.2.0-bin.tar.gz

Lệnh sau đây được sử dụng để xác minh download:

$ ls

Thành công sẽ có trong danh sách:

db-derby-10.4.2.0-bin.tar.gz

### Trích xuất và xác minh Derby

$ tar zxvf db-derby-10.4.2.0-bin.tar.gz

$ ls

Giải nén xong:

db-derby-10.4.2.0-bin db-derby-10.4.2.0-bin.tar.gz

Các lệnh sau đây được sử dụng để sao chép các tập tin từ thư mục giải nén vào thư mục local / usr / derby:

$ su -

passwd:

# cd /home/user

# mv db-derby-10.4.2.0-bin /usr/local/derby

# exit

### 

### Thiết lập môi trường cho Derby

Bạn có thể thiết lập môi trường Derby bằng cách thêm các dòng sau vào **~ / .bashrc** :

export DERBY\_HOME=/usr/local/derby

export PATH=$PATH:$DERBY\_HOME/bin

ApacheHive

18

export CLASSPATH=$CLASSPATH:$DERBY\_HOME/lib/derby.jar:$DERBY\_HOME/lib/derbytools.jar

Sau đó:

$ source ~/.bashrc

### 

### Tạo một thư mục để lưu trữ Metastore

Tạo một thư mục tên data trong thư mục $ DERBY\_HOME để lưu trữ dữ liệu Metastore.

$ mkdir $DERBY\_HOME/data

Cài đặt Derby và thiết lập môi trường đã hoàn tất.

### Bước 7: Cấu hình Metastore của Hive

$ cd $HIVE\_HOME/conf

$ cp hive-default.xml.template hive-site.xml

Chỉnh sửa **hive-site.xml** và thêm các dòng sau đây giữa <configuration> và </ configuration>:

<property>

<name>javax.jdo.option.ConnectionURL</name>

<value>jdbc:derby://localhost:1527/metastore\_db;create=true </value>

<description>JDBC connect string for a JDBC metastore </description>

</property>

Tạo một tập tin có tên jpox.properties và thêm những dòng sau đây vào đó:

javax.jdo.PersistenceManagerFactoryClass=

org.jpox.PersistenceManagerFactoryImpl

org.jpox.autoCreateSchema =false

org.jpox.validateTables =false

org.jpox.validateColumns =false

org.jpox.validateConstraints =false

org.jpox.storeManagerType = rdbms

org.jpox.autoCreateSchema =true

org.jpox.autoStartMechanismMode =checked

org.jpox.transactionIsolation = read\_committed

javax.jdo.option.DetachAllOnCommit=true

javax.jdo.option.NontransactionalRead=true

javax.jdo.option.ConnectionDriverName= org.apache.derby.jdbc.ClientDriver

javax.jdo.option.ConnectionURL= jdbc:derby://hadoop1:1527/metastore\_db;create = true

javax.jdo.option.ConnectionUserName= APP

javax.jdo.option.ConnectionPassword= mine

### 

### Bước 8: Thẩm định cài đặt Hive

Trước khi chạy Hive, bạn cần phải tạo ra các thư mục **tmp/** và thư mục riêng biệt trong Hive HDFS. Bạn cần thiết lập quyền ghi cho các thư mục mới được tạo ra như hình dưới đây:

chmod g+w

Bây giờ đặt chúng trong HDFS trước khi xác minh Hive. Sử dụng các lệnh sau đây:

$ $HADOOP\_HOME/bin/hadoop fs -mkdir /tmp

$ $HADOOP\_HOME/bin/hadoop fs -mkdir /user/hive/warehouse

$ $HADOOP\_HOME/bin/hadoop fs -chmod g+w /tmp

$ $HADOOP\_HOME/bin/hadoop fs -chmod g+w /user/hive/warehouse

Các lệnh sau đây được sử dụng để xác minh cài đặt Hive:

$ cd $HIVE\_HOME

$ bin/hive

Cài đặt thành công của Hive, bạn có thể xem kết quả:

Logging initialized using configuration in jar:file:/home/hadoop/hive-0.9.0/lib/hive-common-0.9.0.jar!/hive-log4j.properties

Hive history file=/tmp/hadoop/hive\_job\_log\_hadoop\_201312121621\_1494929084.txt

………………….

hive>

1. **Thao tác với HIVE**

1. Tạo cơ sở dữ liệu (database) với Hive:

hive> CREATE DATABASE [IF NOT EXISTS] userdb;

Truy vấn xác minh một danh sách database:

hive> SHOW DATABASES;

default

userdb

Xóa database:

hive> DROP DATABASE IF EXISTS userdb CASCADE;

2. Tạo bảng:

hive> CREATE TABLE IF NOT EXISTS employee ( eid int, name String, salary String, destination String)

COMMENT ‘Employee details’

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘\t’

LINES TERMINATED BY ‘\n’

STORED AS TEXTFILE;

Sau khi tạo một bảng trong SQL, chúng ta có thể chèn dữ liệu sử dụng câu lệnh LOAD DATA. (Ta có thể lấy dữ liệu từ file log của hệ thống hoặc file text chứa dữ liệu theo 1 định dạng chuẩn)

Cú pháp như sau:

LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO TABLE tablename

[PARTITION (partcol1=val1, partcol2=val2 ...)]

**Ví dụ:**

Ta sẽ chèn dữ liệu sau vào bảng. Nó là một file văn bản có tên **sample.txt** trong thư mục **/ home / user**.

1201 Gopal 45000 Technical manager

1202 Manisha 45000 Proof reader

1203 Masthanvali 40000 Technical writer

1204 Kiran 40000 Hr Admin

1205 Kranthi 30000 Op Admin

Truy vấn:

hive> LOAD DATA LOCAL INPATH '/home/user/sample.txt'

OVERWRITE INTO TABLE employee;

Xóa bảng:

hive> DROP TABLE IF EXISTS employee;

3. Lấy dữ liệu từ bảng (Select):

Cú pháp của câu truy vấn SELECT như sau:

SELECT [ALL | DISTINCT] select\_expr, select\_expr, ...

FROM table\_reference

[WHERE where\_condition]

[GROUP BY col\_list]

[HAVING having\_condition]

[CLUSTER BY col\_list | [DISTRIBUTE BY col\_list] [SORT BY col\_list]]

[LIMIT number];

**Ví dụ:**

Chúng ta hãy lấy một ví dụ cho SELECT ... WHERE. Giả sử chúng ta có bảng nhân viên như được đưa ra dưới đây, với các trường ID, Name, Salary, Designation, và Dept, tạo một truy vấn để lấy các chi tiết nhân viên có lương hơn 30000.

+------+--------------+-------------+-------------------+--------+

| ID | Name | Salary | Designation | Dept |

+------+--------------+-------------+-------------------+--------+

|1201 | Gopal | 45000 | Technical manager | TP |

|1202 | Manisha | 45000 | Proofreader | PR |

|1203 | Masthanvali | 40000 | Technical writer | TP |

|1204 | Krian | 40000 | Hr Admin | HR |

|1205 | Kranthi | 30000 | Op Admin | Admin |

+------+--------------+-------------+-------------------+--------+

Thực hiện truy vấn:

hive> SELECT \* FROM employee WHERE salary>30000;

Kết quả:

+------+--------------+-------------+-------------------+--------+

| ID | Name | Salary | Designation | Dept |

+------+--------------+-------------+-------------------+--------+

|1201 | Gopal | 45000 | Technical manager | TP |

|1202 | Manisha | 45000 | Proofreader | PR |

|1203 | Masthanvali | 40000 | Technical writer | TP |

|1204 | Krian | 40000 | Hr Admin | HR |

+------+--------------+-------------+-------------------+--------+

Với các truy vấn Select Order By, Select Group By, Select Join thì tương tự (giống với truy vấn trên mysql).

1. **Ứng dụng demo**

Hive không cho insert dữ liệu trực tiếp vào bảng (thông qua file). Các câu lệnh quen thuộc trong SQL như INSERT INTO, DELETE hay UPDATE đều được loại bỏ trong Hive. Hive không hỗ trợ chèn vào bảng hay partition mà ta chỉ có thể overwrite lại toàn bộ dữ liệu bằng lệnh INSERT OVERWRITE TABLE.

Do đó, nhóm em không làm ứng dụng tương tác với Hive như thêm, sửa, xóa nên không có demo được ạ. Ta tương tác với Hive bằng các truy vấn như ở trên vừa giới thiệu. Mong cô thông cảm!

-------------------------------------------END-----------------------------------------------

Nguồn:

<https://www.tutorialspoint.com/hive/index.htm>

<https://bigsonata.wordpress.com>

**PHIẾU ĐÁNH GIÁ CÁC THÀNH VIÊN TRONG NHÓM**

Họ và tên người đánh giá:Nguyễn Thị Phương

Nhóm số:14

*Chú ý: Điền Họ tên của các thành viên còn lại vào từng cột*

|  |  |  |
| --- | --- | --- |
| ***Tiêu chí*** | ***Điểm của thành viên.....Bùi Thị Lan Hương...*** | ***Điểm của thành viên..Nguyễn Thanh Long......*** |
| Lập kế hoạch và phân công nhiệm vụ | 10 | 10 |
| Tạo ra những ý tưởng và giải pháp (x2) | 9 \* 2 | 8 \* 2 |
| Giải quyết các vấn đề được phân công (x3) | 10 \* 3 | 10 \* 3 |
| Giúp thành viên trong nhóm hoàn thành công việc | 10 | 10 |
| Khả năng phối hợp, tương tác với các thành viên khác trong nhóm (x3) | 10 \* 3 | 10 \* 3 |
| **Tổng** | **98/100** | **96/100** |

Thang điểm: Từ 0 đến 10 (rất tốt)

Nhận xét khác (nếu có):.........................................................................................................