

Hadoop数据分析平台 第10周

2012.11.29

【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站


<http://edu.dataguru.cn>

与关系型数据库交换数据

- 文本转换方案
- 自写Java程序
- Sqoop
- 厂商提供的解决方案

Sqoop

- SQL-to-HDFS工具
- 利用JDBC连接关系型数据库
- Sqoop的获取



Index of /sqoop/1.4.2/

../		
sqoop-1.4.2.bin_hadoop-0.20.tar.gz	22-Aug-2012 19:48	4782610
sqoop-1.4.2.bin_hadoop-0.20.tar.gz.asc	22-Aug-2012 19:48	836
sqoop-1.4.2.bin_hadoop-0.20.tar.gz.md5sum	22-Aug-2012 19:48	70
sqoop-1.4.2.bin_hadoop-0.23.tar.gz	22-Aug-2012 19:48	4783221
sqoop-1.4.2.bin_hadoop-0.23.tar.gz.asc	22-Aug-2012 19:48	836
sqoop-1.4.2.bin_hadoop-0.23.tar.gz.md5sum	22-Aug-2012 19:48	70
sqoop-1.4.2.bin_hadoop-1.0.0.tar.gz	22-Aug-2012 19:48	4782922
sqoop-1.4.2.bin_hadoop-1.0.0.tar.gz.asc	22-Aug-2012 19:48	836
sqoop-1.4.2.bin_hadoop-1.0.0.tar.gz.md5sum	22-Aug-2012 19:48	71
sqoop-1.4.2.bin_hadoop-2.0.0-alpha.tar.gz	22-Aug-2012 19:48	4785174
sqoop-1.4.2.bin_hadoop-2.0.0-alpha.tar.gz.asc	22-Aug-2012 19:48	836
sqoop-1.4.2.bin_hadoop-2.0.0-alpha.tar.gz.md5sum	22-Aug-2012 19:48	77
sqoop-1.4.2.tar.gz	22-Aug-2012 19:48	746101
sqoop-1.4.2.tar.gz.asc	22-Aug-2012 19:48	836
sqoop-1.4.2.tar.gz.md5sum	22-Aug-2012 19:48	53

2012.11.29

Hadoop-0.20.2下使用Sqoop

- SQOOP不支持此版本，可使用CDH3。也可以通过拷贝相应的包到sqoop-1.2.0-CDH3B4/lib下，依然可以使用。

- CDH3和SQOOP 1.2.0的下载地址

<http://archive.cloudera.com/cdh/3/hadoop-0.20.2-CDH3B4.tar.gz>

<http://archive.cloudera.com/cdh/3/sqoop-1.2.0-CDH3B4.tar.gz>

- 其中sqoop-1.2.0-CDH3B4依赖hadoop-core-0.20.2-CDH3B4.jar，所以你需要下载hadoop- 0.20.2-CDH3B4.tar.gz，解压缩后将hadoop-0.20.2-CDH3B4/hadoop-core-0.20.2- CDH3B4.jar复制到sqoop-1.2.0-CDH3B4/lib中。
- 另外，sqoop导入mysql数据运行过程中依赖mysql-connector-java-*.jar，所以你需要下载mysql-connector-java-*.jar并复制到sqoop-1.2.0-CDH3B4/lib中。

- 修改SQOOP的文件configure-sqoop , 注释掉hbase和zookeeper检查 (除非你准备使用HABASE等HADOOP上的组件)

```
#if [ ! -d "${HBASE_HOME}" ]; then
# echo "Error: $HBASE_HOME does not exist!"
# echo 'Please set $HBASE_HOME to the root of your HBase installation.'
# exit 1
#fi

#if [ ! -d "${ZOOKEEPER_HOME}" ]; then
# echo "Error: $ZOOKEEPER_HOME does not exist!"
# echo 'Please set $ZOOKEEPER_HOME to the root of your ZooKeeper installation.'
# exit 1
#fi
```

- 3、启动HADOOP , 配置好相关环境变量 (例如\$HADOOP_HOME) , 就可以使用SQOOP了

```
% sqoop help
usage: sqoop COMMAND [ARGS]
```

Available commands:

codegen	Generate code to interact with database records
create-hive-table	Import a table definition into Hive
eval	Evaluate a SQL statement and display the results
export	Export an HDFS directory to a database table
help	List available commands
import	Import a table from a database to HDFS
import-all-tables	Import tables from a database to HDFS
list-databases	List available databases on a server
list-tables	List available tables in a database
version	Display version information

See 'sqoop help COMMAND' for information on a specific command.

import

```
% sqoop help import
```

```
usage: sqoop import [GENERIC-ARGS] [TOOL-ARGS]
```

```
Common arguments:
```

<code>--connect <jdbc-uri></code>	Specify JDBC connect string
<code>--driver <class-name></code>	Manually specify JDBC driver class to use
<code>--hadoop-home <dir></code>	Override \$HADOOP_HOME
<code>--help</code>	Print usage instructions

<code>-P</code>	Read password from console
<code>--password <password></code>	Set authentication password
<code>--username <username></code>	Set authentication username
<code>--verbose</code>	Print more information while working
<code>...</code>	

从mysql导入数据的例子

```
% sqoop import --connect jdbc:mysql://localhost/hadoopguide \  
> --table widgets -m 1  
10/06/23 14:44:18 INFO tool.CodeGenTool: Beginning code generation  
...  
10/06/23 14:44:20 INFO mapred.JobClient: Running job: job_201006231439_0002  
10/06/23 14:44:21 INFO mapred.JobClient: map 0% reduce 0%  
10/06/23 14:44:32 INFO mapred.JobClient: map 100% reduce 0%  
10/06/23 14:44:34 INFO mapred.JobClient: Job complete:  
job_201006231439_0002  
...  
10/06/23 14:44:34 INFO mapreduce.ImportJobBase: Retrieved 3 records.  
  
% hadoop fs -cat widgets/part-m-00000  
1,sprocket,0.25,2010-02-10,1,Connects two gizmos  
2,gizmo,4.00,2009-11-30,4,null  
3,gadget,99.99,1983-08-13,13,Our flagship product
```

导入到Hbase的命令

```
sqoop import --connect jdbc:mysql://mysqlserver_IP/databaseName --table
  datatable --hbase-create-table --hbase-table hbase_tablename --column-
  family col_fam_name --hbase-row-key key_col_name
```

- 其中，databaseName 和datatable 是mysql的数据库和表名，hbase_tablename是要导成hbase的表名，key_col_name可以指定datatable中哪一列作为hbase新表的rowkey，col_fam_name是除rowkey之外的所有列的列族名

- sqoop从oracle导入，需要有ojdbc6.jar,放在\$SQOOP_HOME/lib里，不用添加到classpath里，因为sqoop会自己遍历lib文件夹并添加里面的所有jar包 --connect与mysql的不一样，如下（shell脚本中的主要部分）

#Oracle的连接字符串，其中包含了Oracle的地址，SID，和端口

CONNECTURL=jdbc:oracle:thin:@172.7.10.16:1521:orcl

#使用的用户名

ORACLENAME=scott

#使用的密码

ORACLEPASSWORD=wang123456

#需要从Oracle中导入的表名

oracleTableName=test

#需要从Oracle中导入的表中的字段名

columns=ID,STATE

#将Oracle中的数据导入到HDFS后的存放路径

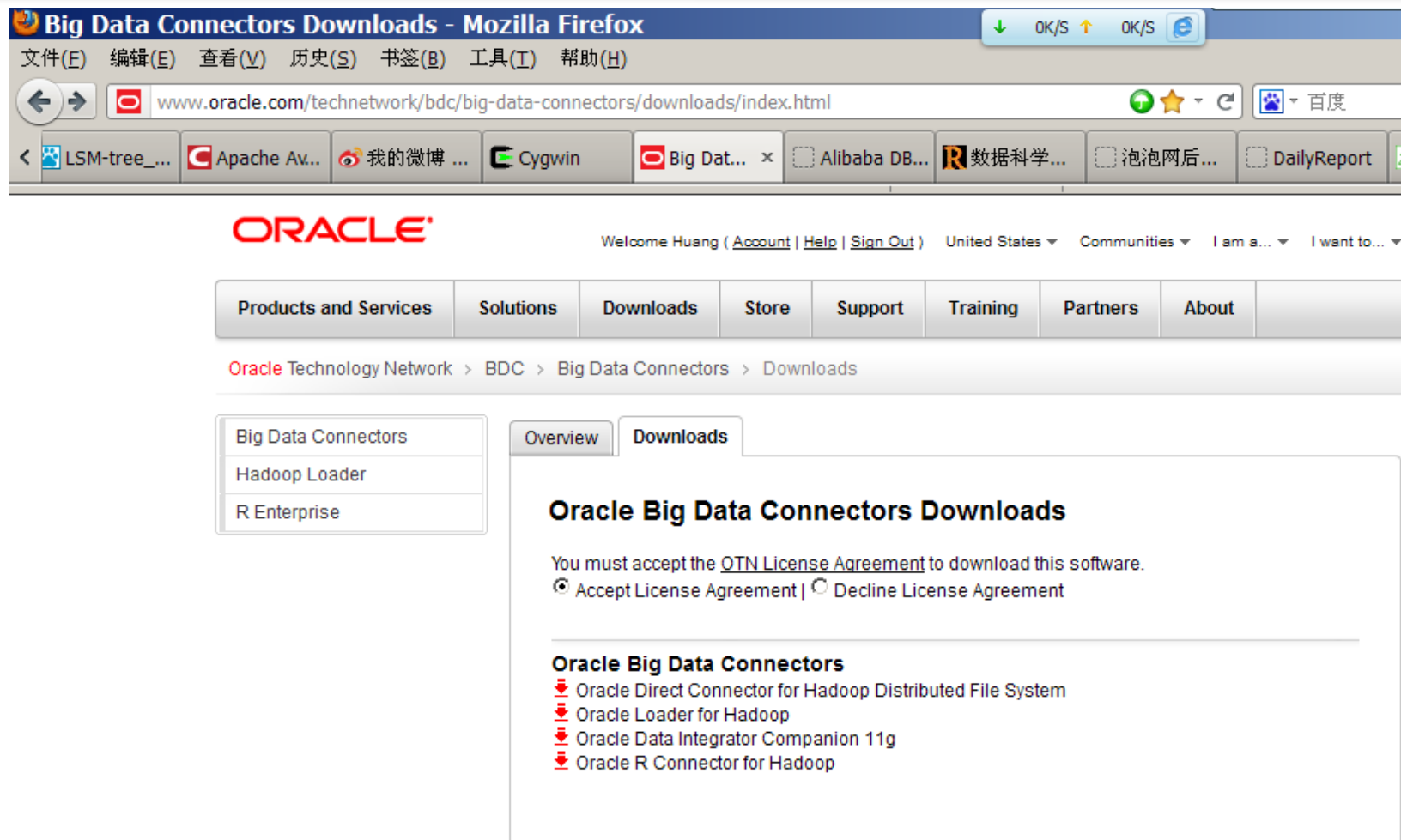
#hdfsPath=/tmp/

2012.11.29

#执行导入逻辑。将Oracle中的数据导入到HDFS中

```
sqoop import --append --connect $CONNECTURL --username $ORACLENAME  
--password $ORACLEPASSWORD --m 1 --table $oracleTableName --columns  
$columns --hbase-create-table --hbase-table or1 --hbase-row-key STATE --  
column-family or1
```

Oracle Big Data Connectors



The screenshot shows a Mozilla Firefox browser window with the title "Big Data Connectors Downloads - Mozilla Firefox". The address bar displays the URL: www.oracle.com/technetwork/bdc/big-data-connectors/downloads/index.html. The browser's menu bar includes "文件(E)", "编辑(E)", "查看(V)", "历史(S)", "书签(B)", "工具(T)", and "帮助(H)". The toolbar shows navigation buttons, a search bar with "百度", and several open tabs including "LSM-tree...", "Apache Av...", "我的微博...", "Cygwin", "Big Dat...", "Alibaba DB...", "数据科学...", "泡泡网后...", "DailyReport", and a green icon.

The Oracle website header features the "ORACLE" logo in red, followed by a welcome message "Welcome Huang (Account | Help | Sign Out)", and links for "United States", "Communities", "I am a...", and "I want to...". A navigation menu contains "Products and Services", "Solutions", "Downloads", "Store", "Support", "Training", "Partners", and "About".

The breadcrumb trail reads: "Oracle Technology Network > BDC > Big Data Connectors > Downloads".

On the left, a sidebar lists "Big Data Connectors", "Hadoop Loader", and "R Enterprise". The main content area has two tabs: "Overview" and "Downloads". The "Downloads" tab is active, displaying the title "Oracle Big Data Connectors Downloads".

Below the title, a message states: "You must accept the [OTN License Agreement](#) to download this software." Below this, there are two radio buttons: "Accept License Agreement" (selected) and "Decline License Agreement".

A section titled "Oracle Big Data Connectors" lists the following connectors with red download icons:

- Oracle Direct Connector for Hadoop Distributed File System
- Oracle Loader for Hadoop
- Oracle Data Integrator Companion 11g
- Oracle R Connector for Hadoop

2012.11.29

Oracle HDFS直接连接器（ODCH）实验

- 实验 1: 直接访问单个 HDFS 文件
- 步骤1: 配置操作系统的目录和数据库的Directory对象
- 步骤2: 创建外部表
- 步骤3: 在Hadoop中放入示例文件
- 步骤4: 生成 “位置文件”
- 步骤5: 检查结果
- 步骤6: 改动HDFS文件，检查结果.

- **软件环境:** 本实验主要由以下软件搭建而成：Oracle Enterprise Linux, Oracle 11g, Java SE6pdate30, Apache Hadoop, Oracle Connector for Hadoop等。
- **实验用到的文件:** 实验用到的文件保存在 /home/hadoop/training/ODCH 底下. 包括脚本文件以及一些示例数据文件。
- **环境变量:** 在文件olhodchenv.sh中保存实验中需要用到环境变量. 为了简化操作，已经在实验中的\$HOME/.bash_profile引用该文件，这些环境变量会自动生效。

变量名	变量值
ORACLE_HOME	/home/oracle/app/oracle/product/11.2.0/dbhome_2
HADOOP_HOME	/opt/hadoop
DIRECTHDFS_HOME	/opt/ODCH
ORAHDFS_JAR	\$DIRECTHDFS_HOME /jlib/orahdfs.jar
HDFS_BIN_PATH	\$DIRECTHDFS_HOME /bin
HADOOP_CONF_DIR	\${HADOOP_HOME}/conf
ORACLE_SID	orcl

2012.11.29

下表中也列出了实验中可能需要的一些信息。

项目	值
虚拟机 IP	172.16.22.131
虚拟机主机名	bigdata01
Hadoop default FS	hdfs://bigdata01:9000
Hadoop Job Tracker URL	hdfs://bigdata01:9001
实验用操作系统用户密码	hadoop/oracle
实验用数据库用户密码	Scott/tiger
操作系统oracle用户密码	oracle/oracle

实验 1: 直接访问HDFS数据文件

- Oracle的HDFS直接连接器允许从数据库中直接访问HDFS的数据文件。支持的数据文件格式取决于ORACLE_LOADER的驱动程序。
- 在实验1里, 我们将会直接访问HDFS上的几个带分割符的文本文件。我们可以在数据库中用SQL来查询该文件。

步骤1

步骤1: 配置hdfs_stream script文件。在使用直接连接器前，需要配置hdfs_stream 脚本。

hdfs_stream 是 包含在ODCH的安装包中(ODCH_HOME/bin). 我们需要在脚本中指定HADOOP_HOME和DIRECTHDFS_HOME.

```
PROMPT> cd /home/hadoop/training/ODCH
```

```
PROMPT> vi ${DIRECTHDFS_HOME}/bin/hdfs_stream
```

```
...
```

```
export HADOOP_HOME=/opt/hadoop
```

```
...
```

```
export DIRECTHDFS_HOME=/opt/ODCH
```

```
...
```

另外Oracle用户需要在 \${DIRECTHDFS_LOG_DIR} 目录中创建log/bad文件. 所以要确保Oracle用户有读写权限.

```
PROMPT> su - oracle
```

```
PROMPT> touch /opt/ODCH/log/oracle_access_test
```

```
PROMPT> rm /opt/ODCH/log/oracle_access_test
```

■ 配置操作系统的目录和数据库的Directory对象

在ODCH里面，需要用到3个Directory对象：

HDFS_BIN_PATH: hdfs_stream脚本所在目录。

XTAB_DATA_DIR：用来存放“位置文件” (location files)的目录。“位置文件” (location files) 是一个配置文件，里面包含HDFS的文件路径/文件名以及文件编码格式。

ODCH_LOG_DIR, Oracle用来存放外部表的log/bad等文件的目录。

对于第一个目录，已经在操作系统存在。对于第二和第三个目录，我们将会在操作系统中新创建，并且授予oracle用户读写权限。

检查脚本文件并运行之：

```
PROMPT> cat lab4.2_setup_os_dir.sh
```

```
mkdir -p /home/hadoop/training/ODCH/logs
```

```
mkdir -p /home/hadoop/training/ODCH/extdir
```

```
chmod 777 /home/hadoop/training/ODCH/logs
```

```
chmod 777 /home/hadoop/training/ODCH/extdir
```

```
PROMPT> ./lab4.2_setup_os_dir.sh
```

步骤2续

连接到数据库，建立相应的3个Directory对象,以及相关授权。

```
PROMPT> sqlplus 'sys/oracle as sysdba'
```

检查脚本文件并运行之:

```
SQL> !cat lab4.2_setup_DB_dir.sql
```

```
SET ECHO ON
```

```
create or replace directory ODCH_LOG_DIR as '/home/hadoop/training/ODCH/logs';
```

```
grant read, write on directory ODCH_LOG_DIR to SCOTT;
```

```
create or replace directory ODCH_DATA_DIR as '/home/hadoop/training/ODCH/extdir';
```

```
grant read, write on directory ODCH_DATA_DIR to SCOTT;
```

```
create or replace directory HDFS_BIN_PATH as '/opt/ODCH/bin';
```

```
grant execute on directory HDFS_BIN_PATH to SCOTT;
```

```
SQL> @lab4.2_setup_DB_dir.sql
```

步骤3: 创建外部表

我们将会创建外部表，里面有个ODCH的关键参数-- “preprocessor HDFS_BIN_PATH:hdfs_stream” 。

另外，下面SQL脚本中的LOCATION对应的文件不用预先存在，我们会在步骤4中生成。

在LOCATION中使用多个文件，可以使Oracle可以多个程序并行访问HDFS。

```
PROMPT> sqlplus scott/tiger
```

检查脚本文件并运行之:

```
SQL> !cat lab4.3_ext_tab.sql
```

```
drop table odch_ext_table;
```

```
CREATE TABLE odch_ext_table
```

```
( ID NUMBER
```

```
,OWNER VARCHAR2(128)
```

```
,NAME VARCHAR2(128)
```

```
,MODIFIED DATE
```

```
,Val NUMBER
```

```
) ORGANIZATION EXTERNAL
```

```
(TYPE oracle_loader
```

```
DEFAULT DIRECTORY "ODCH_DATA_DIR"
```

ACCESS PARAMETERS

(

records delimited by newline

preprocessor HDFS_BIN_PATH:hdfs_stream

badfile ODCH_LOG_DIR:'odch_ext_table%a_%p.bad'

logfile ODCH_LOG_DIR:'odch_ext_table%a_%p.log'

fields terminated by ','

missing field values are null

(

ID DECIMAL EXTERNAL,

OWNER CHAR(200),

步骤3续

```
NAME CHAR(200),  
    MODIFIED CHAR DATE_FORMAT DATE MASK "YYYY-MM-DD HH24:MI:SS",  
    Val DECIMAL EXTERNAL  
)  
  
)  
LOCATION (  
'odch_ext_table1.loc' ,  
'odch_ext_table2.loc' ,  
'odch_ext_table3.loc' ,  
'odch_ext_table4.loc'  
)  
)  
PARALLEL REJECT LIMIT UNLIMITED;;  
SQL> @lab4.3_ext_tab.sql
```

2012.11.29

步骤4: 在Hadoop中放入示例文件

ODCH从Hadoop文件系统中读取数据. 所以我们要先在Hadoop中放入几个的数据文件.

下面的脚本先在Hadoop中建立一个目录, 然后把odch*.dat放入该目录中.

检查脚本文件并运行之:

```
PROMPT> cat lab4.4_hdfs_setup.sh
```

```
${HADOOP_HOME}/bin/hadoop fs -rmr odch_data
```

```
${HADOOP_HOME}/bin/hadoop fs -mkdir odch_data
```

```
${HADOOP_HOME}/bin/hadoop fs -put odch*.dat odch_data
```

```
echo "rows in file:"
```

```
wc -l odch*.dat
```

```
PROMPT> ./lab4.4_hdfs_setup.sh
```


步骤5: 生成 “位置文件”

我们需要让Oracle Hadoop直接连接器知道需要访问的HDFS上的文件路径。下面运行的程序将会生成包含HDFS上文件路径的“位置文件”。

检查脚本文件并运行之

```
PROMPT>cat lab4.5_create_loc_file.sh
```

```
hadoop jar \
```

```
  ${ORAHDFS_JAR} oracle.hadoop.hdfs.exttab.ExternalTable \
```

```
  -D oracle.hadoop.hdfs.exttab.tableName=odch_ext_table \
```

```
  -D oracle.hadoop.hdfs.exttab.datasetPaths=odch_data \
```

```
  -D oracle.hadoop.hdfs.exttab.datasetRegex=odch*.dat \
```

```
  -D oracle.hadoop.hdfs.exttab.connection.url="jdbc:oracle:thin:@//172.16.22.131:1521/orcl" \
```

```
  -D oracle.hadoop.hdfs.exttab.connection.user=SCOTT \
```

```
  -publish
```

```
PROMPT> ./lab4.5_create_loc_file.sh
```

需要输入数据库用户的密码，本实验中是 ‘tiger’ 。

步骤5续

检查位置文件内容.

```
PROMPT> cat /home/hadoop/training/ODCH/extdir/odch_ext_table*.loc
```

```
CompressionCodec=
```

```
hdfs://bigdata01:9000/user/hadoop/odch_data/odch.dat
```

这里 CompressionCodec 是默认值, HDFS 文件指向

```
hdfs://bigdata01:9000/user/hadoop/odch_data/odch.dat
```

步骤6: 检查结果

```
PROMPT> sqlplus scott/tiger
```

```
SQL> select count(*) from odch_ext_table;
```

```
90000
```

91000是符合odch.*.dat的文件的总行数.

我们可以在sqlplus中设置 autotrace , 看看执行计划中是否有并行操作(“PX”)出现.

```
SQL> set autotrace trace exp
```

```
SQL> select count(*) from odch_ext_para_table;
```

步骤6续

Execution Plan

Plan hash value: 2012719727

Id	Operation	Name	Rows	Cost (%CPU)	Time	TQ	IN-OUT	PQ Distrib
0	SELECT STATEMENT		1	16(0)	00:00:01			
1	SORT AGGREGATE		1					
2	PX COORDINATOR							
3	PX SEND QC (RANDOM)	:TQ10000	1		Q1,00	P->S	QC (RAND)	
4	SORT AGGREGATE		1		Q1,00	PCWP		
5	PX BLOCK ITERATOR		8168	16 (0)	00:00:01	Q1,00	PCWC	
6	EXTERNAL TABLE ACCESS FULL	ODCH_EXT_PARA_TABLE	8168	16 (0)	00:00:01	Q1,00	PCWP	

当然，我们也可以进行其他的SQL语句，比如join, where, group之类的。我们也可以通过Create Table As Select方式将数据完全装载到数据库中。

2012.11.29

步骤7

删除部分文件，从数据库中检查结果:

```
PROMPT> hadoop fs -rm odch_data/odch1.dat
```

```
SQL> select count(*) from odch_ext_para_table;
```

```
41000
```

数据已经更新。

Oracle Hadoop装载程序

实验: 装载Hadoop文件到数据库

步骤1: 创建目标表

步骤2: 在Hadoop中放入示例文件

步骤3: 运行Oracle Hadoop装载程序

步骤4: 验证结果

- **软件环境:** 本实验主要由以下软件搭建而成：Oracle Enterprise Linux, Oracle 11g, Java SE6update30, Apache Hadoop, Oracle Connector for Hadoop等。
- **实验用到的文件:** 实验用到的文件保存在 /home/hadoop/training/OLH 目录下，包括脚本文件以及一些示例数据文件。
- **环境变量:** 在文件olhodchenv.sh中保存了实验中需要用到的环境变量。为了简化操作，已经在实验中的\$HOME/.bash_profile引用该文件，这些环境变量会自动生效。

变量名	变量值
ORACLE_HOME	/home/oracle/app/oracle/product/11.2.0/dbhome_2
HADOOP_HOME	/opt/hadoop
OLH_HOME	/opt/OLH
OLH_JAR	OLH_HOME/jlib/oraloader.jar
HADOOP_CONF_DIR	\${HADOOP_HOME}/conf
ORACLE_SID	orcl

下表中也列出了实验中可能需要的一些信息.

项目	值
虚拟机 IP	172.16.22.131
虚拟机主机名	bigdata01
Hadoop default FS	hdfs://bigdata01:9000
Hadoop Job Tracker URL	hdfs://bigdata01:9001
实验用操作系统用户密码	hadoop/oracle
实验用数据库用户密码	Scott/tiger
操作系统oracle用户密码	oracle/oracle
数据库超级用户	sys/oracle

检查环境变量是否正确设置.

```
PROMPT> env
```

应该能看到上面提到的环境变量.

检查hadoop是否正常.

```
PROMPT> hadoop dfsadmin -report
```

检查数据库是否正常

```
PROMPT> sqlplus scott/tiger
```

```
SQL> select * from tab;
```

步骤1: 创建目标表

在实验里, 我们将会把一个Hadoop文件系统上的文件装载到数据库中 (使用JDBC 连接)。
这是OLH的最基本功能。首先, 我们在数据库中新建一个表, 我们的数据将会装载到这个表里. 检查脚本文件并运行之:

```
PROMPT> cd /home/hadoop/training/OLH
```

```
PROMPT> sqlplus scott/tiger
```

```
SQL> !cat lab1.1_target_tab.sql
```

```
-- Drop table if table exists
```

```
drop table olh_table purge;
```

```
-- create table olh_table (col1 NUMBER, col2 VARCHAR2(30), col3 VARCHAR2(128), col4  
date);
```

```
create table olh_table(
```

```
col1 NUMBER, col2 VARCHAR2(30), col3 VARCHAR2(128), col4 date );
```

```
SQL> @lab1.1_target_tab.sql;
```

步骤2: 在Hadoop中放入示例文件

因为OLH需要从Hadoop文件系统中读取数据，所以我们先要在Hadoop中放入一个的数据文件。下面的脚本先在Hadoop中建立一个目录，然后把data.dat放入该目录中。

检查脚本文件并运行之：

```
PROMPT> cat ./lab1.2_init_hadoop_files.sh
```

```
#Set up input directory
```

```
hadoop fs -rmr olh_lab_in
```

```
hadoop fs -mkdir olh_lab_in
```

```
hadoop fs -put olh_lab.dat olh_lab_in/data.dat
```

```
PROMPT> ./lab1.2_init_hadoop_files.sh
```

步骤3: 运行Oracle Hadoop装载程序

我们现在就可以开始装载程序了.

检查脚本文件:

```
PROMPT> cat ./lab1.3_run_loader.sh
```

```
hadoop fs -rmr olh_lab_out
```

```
hadoop jar $OLH_JAR oracle.hadoop.loader.OraLoader -conf MyConf.xml
```

在装载程序中，需要建立一个在Hadoop中新建一个目录，用来存放“_SUCCESS”和“_logs”文件；在使用离线装载选项时，还有可能需要存放一些离线装载的文件。所以，我们先要确保没有该目录，以免创建失败。

装载程序需要读取一些配置信息，在这个例子中，我们使用配置文件: MyConf.xml. 仔细检查 MyConf.xml. 文件里包含里一些运行OLH所需的主要参数，如下表所示：

步骤3续

mapreduce.inputformat.class	指定输入文件的格式。除了文本文件，还支持hive格式文件。也可以是自定义的文件格式。
mapred.input.dir	Hadoop里输入的数据文件（含路径）
mapreduce.outputformat.class	指定装载的输出方式 在线装载： OCIOutputFormat(*), JDBCOutputFormat 离线装载： DataPumptOutputFormat , DelimitedTextOutputFormat
mapred.output.dir	输出目录（同时也是LOG所在目录）
oracle.hadoop.loader.loaderMapFile	文件与表的对应关系，包括表名，栏位对应等
oracle.hadoop.loader.connection.url/user/pass	目标数据库的连接信息，包括url，用户名，密码

(注: OCIOutputFormat 只支持 64 位的 Linux)

对于实验, 最关键参数是`mapreduce.outputformat.class`, 确保它的值是
`JDBCOutputFormat`.

运行脚本文件.

```
PROMPT>./lab1.3_run_loader.sh
```

除了使用`-conf=配置文件`, 我们也可以用 “`-D参数=值`” 的方式来传递参数, 而且, `-D` 方式会覆盖`-conf`设定的值.

比如:

```
hadoop jar ${OLH_JAR} oracle.hadoop.loader.OraLoader -D mapred.input.dir  
olh_lab_in -D mapreduce.inputformat.class  
oracle.hadoop.loader.lib.input.DelimitedTextInputFormat
```

步骤4: 验证结果

登录到数据库，检查数据是否已经成功进入到数据库

```
PROMPT> sqlplus scott/tiger
```

```
SQL> select count(*) from olh_table;
```

```
10000
```

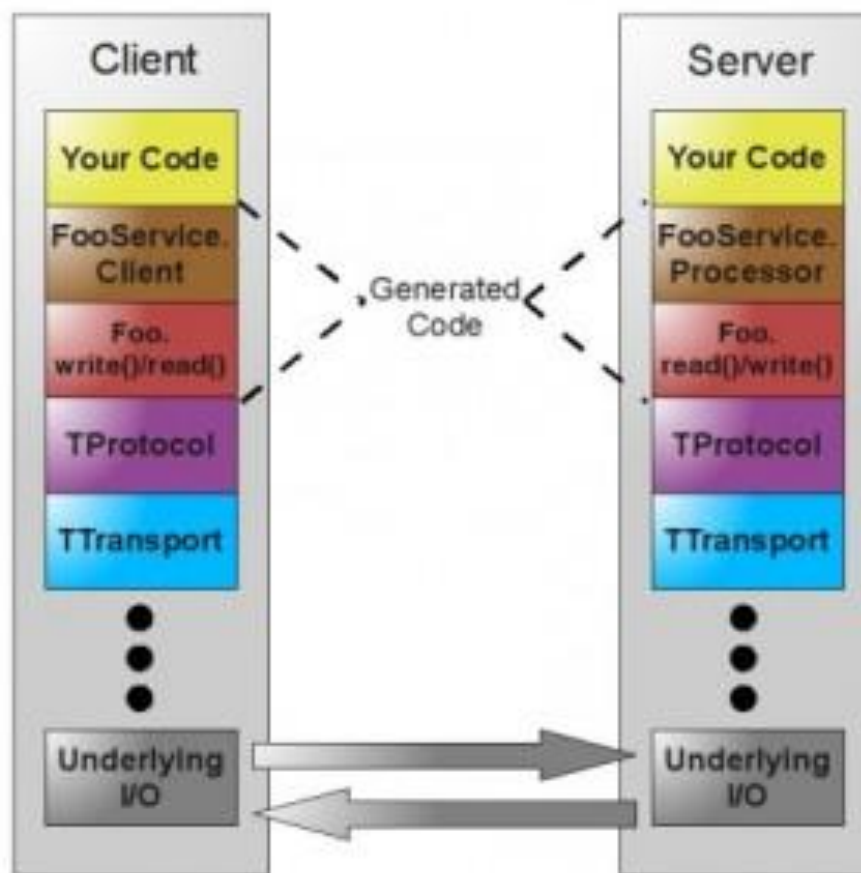
返回10000，表示我们已经成功装载了10000行记录到数据库里面了。

其它可以尝试的实验

- 装载多个Hadoop文件
- 将Hadoop文件装载到datapump格式文件
- 将Hadoop文件装载到预分区的datapump格式文件

应用与Hbase的对接：通过Thrift

- Thrift是一个跨语言的服务部署框架，最初由Facebook于2007年开发，2008年进入Apache开源项目。Thrift通过一个中间语言 (IDL, 接口定义语言)来定义RPC的接口和数据类型，然后通过一个编译器生成不同语言的代码（目前支持C++,Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, Smalltalk和OCaml），并由生成的代码负责RPC协议层和传输层的实现。



2012.11.29

- Thrift框架介绍：<http://dongxicheng.org/search-engine/thrift-framework-intro/>
- Thrift使用指南：<http://dongxicheng.org/search-engine/thrift-guide/>

PHP通过Thrift连接Hbase的主要步骤

- 下载并且编译、安装Thrift
- 生成php和hbase的接口文件
- 把PHP客户端需要的包及刚才生成的接口文件复制出来供php程序调用
- 启动hbase thrift server , 测试php连接hbase
- 参考文档：<http://www.it165.net/pro/html/201206/2827.html>

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 <http://edu.dataguru.cn>



Thanks

FAQ时间