小象科技
ChinaHadoop.cn

数据分析系统Hive

讲师：董西成

主要内容

小象科技 ChinaHadoop.cn

1. Hive背景及应用场景
2. Hive基本架构
3. Hive使用方式
4. HQL查询语句
5. Hive总结及其类似开源系统

主要内容

小象科技 ChinaHadoop.cn

1. Hive背景及应用场景
2. Hive基本架构
3. Hive使用方式
4. HQL查询语句
5. Hive总结及其类似开源系统

Hive是什么？(以wordcount为例)

小象科技 ChinaHadoop.cn

```
public static class TokenizerMapper
     extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Context context
                  ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}
```

```
public static class IntSumReducer extends
        Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
            Context context) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args)
            .getRemainingArgs();
    if (otherArgs.length != 2) {
        System.err.println("Usage: wordcount <in> <out>");
        System.exit(2);
    }
    Job job = new Job(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

SELECT word, COUNT(*) FROM doc LATERAL VIEW explode(split(text, ' ')) lTable as word GROUP BY word;

Hive是什么？

小象科技 ChinaHadoop.cn

- 由facebook开源，最初用于解决海量结构化的日志数据统计问题；
  - ✓ **ETL（Extraction-Transformation-Loading）工具**
- 构建在Hadoop之上的数据仓库；
  - ✓ **数据计算使用MR，数据存储使用HDFS**
- Hive 定义了一种类 SQL 查询语言——HQL；
  - ✓ **类似SQL，但不完全相同**
- 通常用于进行离线数据处理（采用MapReduce）；
- 可认为是一个HQL→MR的语言翻译器。

Hive典型应用场景

小象科技 ChinaHadoop.cn

➢ 日志分析

- ✓统计网站一个时间段内的pv、uv
- ✓多维度数据分析
- ✓大部分互联网公司使用Hive进行日志分析，包括百度、淘宝等

➢ 其他场景

- ✓海量结构化数据离线分析
- ✓低成本进行数据分析（不直接编写MR）

为什么使用Hive？

小象科技 ChinaHadoop.cn

- 简单，容易上手
  - ✓ **提供了类SQL查询语言HQL；**
- 为超大数据集设计的计算/扩展能力
  - ✓ **MR作为计算引擎，HDFS作为存储系统**
- 统一的元数据管理（HCalalog）
  - ✓ **可与Pig、Presto等共享**

有了Hive，还需要自己写MR程序吗？

小象科技 ChinaHadoop.cn

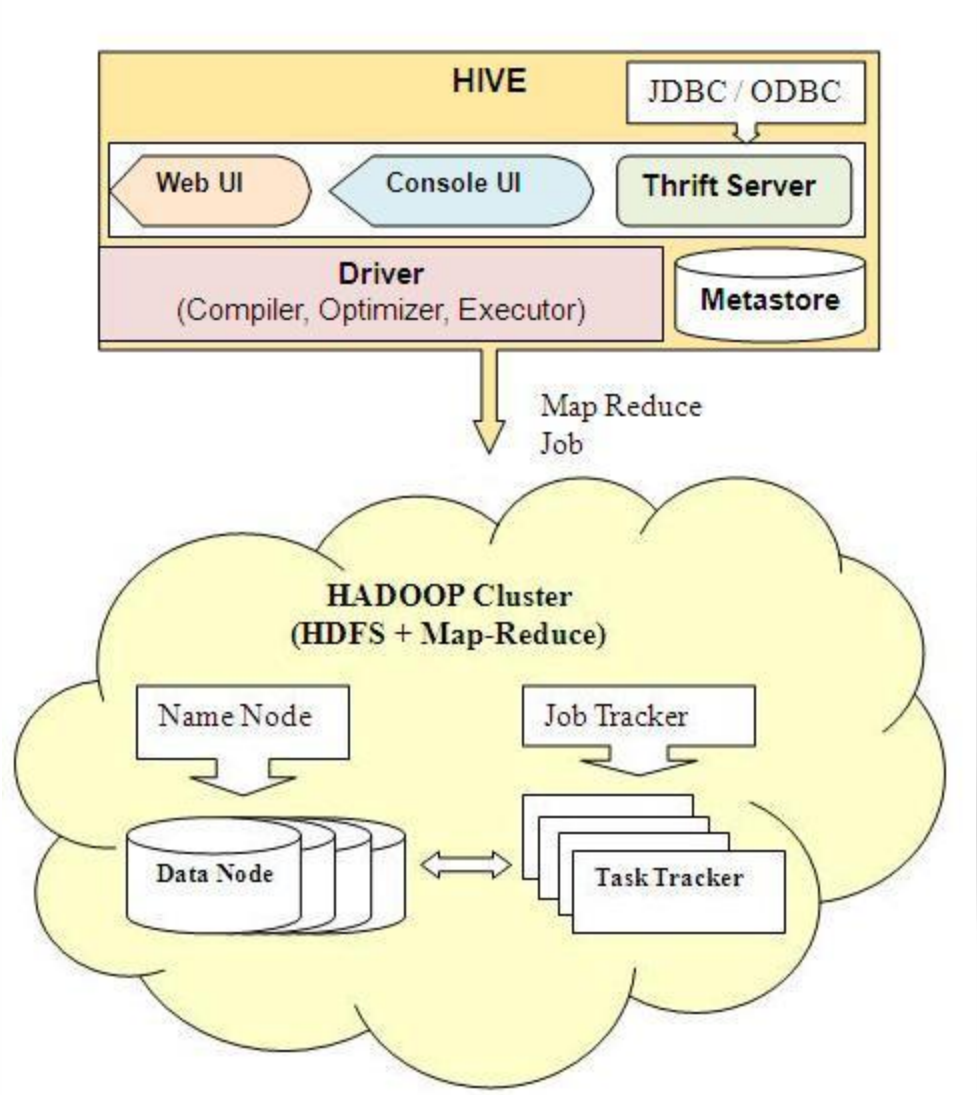
- Hive的HQL表达的能力有限
  - ✓迭代式算法无法表达
  - ✓有些复杂运算用HQL不易表达
- Hive效率较低
  - ✓Hive自动生成MapReduce作业，通常不够智能；
  - ✓HQL调优困难，粒度较粗
  - ✓可控性差

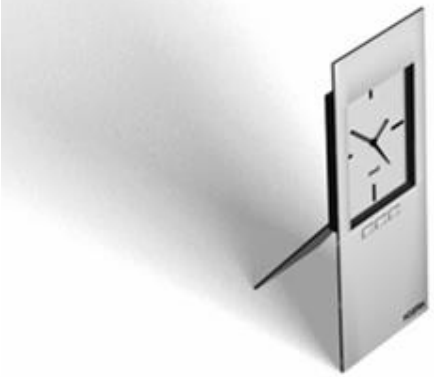
主要内容

小象科技 ChinaHadoop.cn

1. Hive背景及应用场景
2. Hive基本架构
3. Hive使用方式
4. HQL查询语句
5. Hive总结及其类似开源系统

Hive基本架构

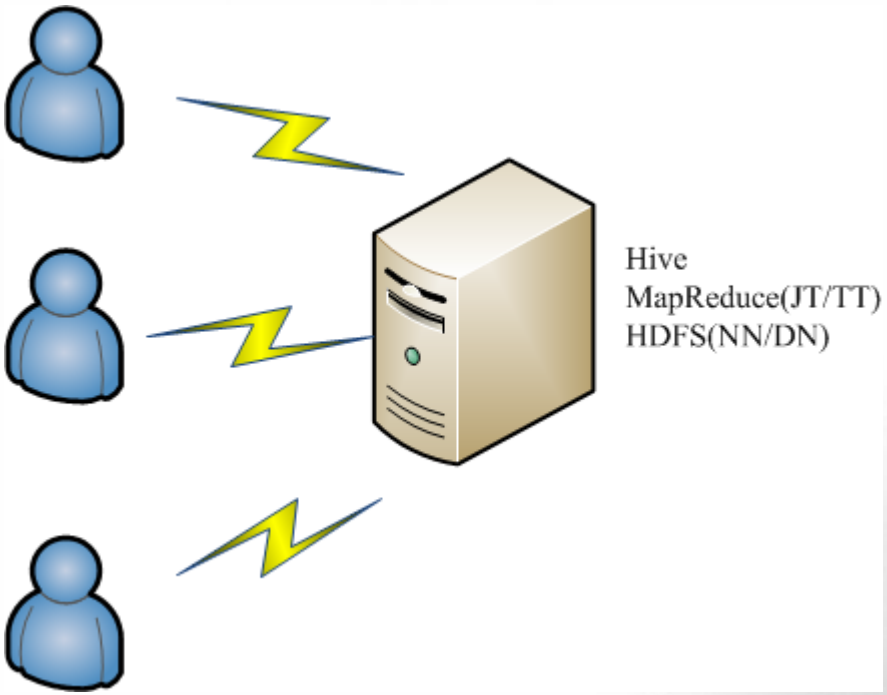

HIVE
JDBC / ODBC
Web UI
Console UI
Thrift Server
Driver
(Compiler, Optimizer, Executor)
Metastore
Map Reduce
Job
HADOOP Cluster
(HDFS + Map-Reduce)
Name Node
Job Tracker
Data Node
Task Tracker



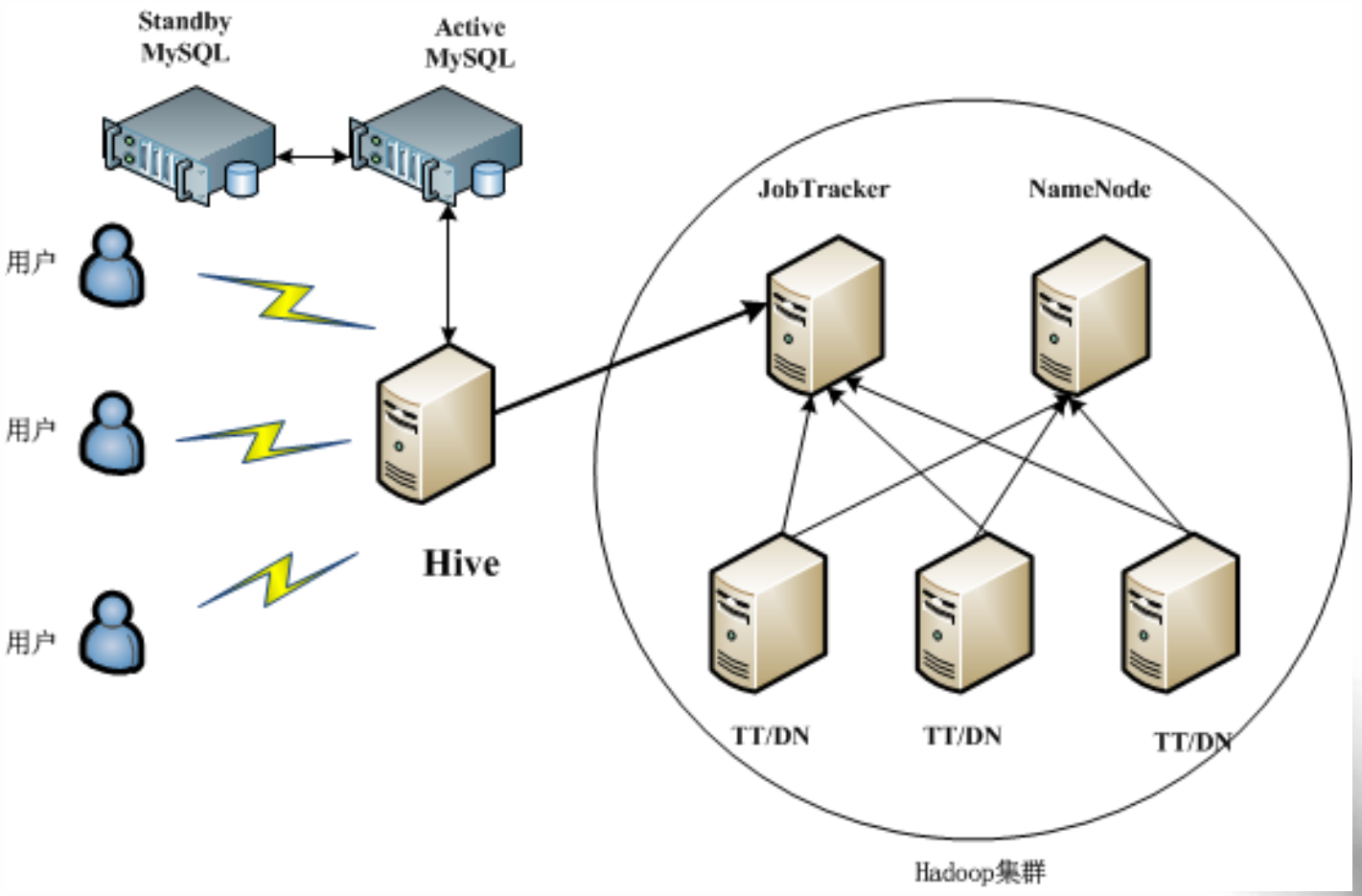
Hive各模块组成

- 用户接口
  - ✓ **包括 CLI，JDBC/ODBC，WebUI**
- 元数据存储（metastore）
  - ✓ **默认存储在自带的数据库derby中，线上使用时一般换为MySQL**
- 驱动器（Driver）
  - ✓ **解释器、编译器、优化器、执行器**
- Hadoop
  - ✓ **用 MapReduce 进行计算，用 HDFS 进行存储**

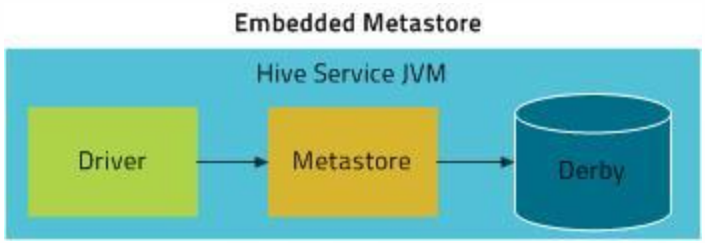
Hive部署架构-实验环境

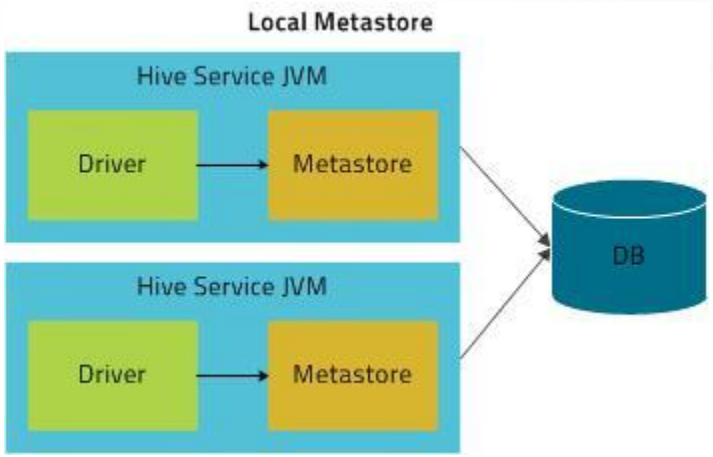


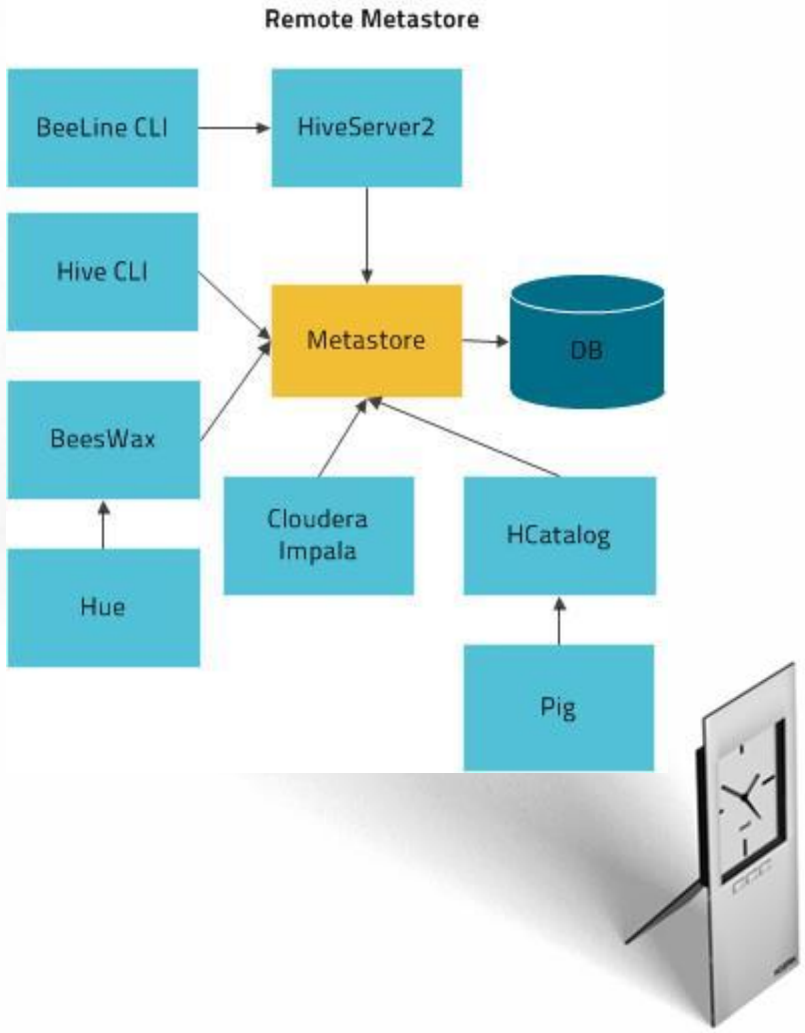
Hive部署架构-生产环境



Hive部署架构-metastore服务







主要内容

小象科技 ChinaHadoop.cn

1. Hive背景及应用场景
2. Hive基本架构
3. Hive使用方式
4. HQL查询语句
5. Hive总结及其类似开源系统

CLI（Command Line Interface）

小象科技 ChinaHadoop.cn

${HIVE_HOME}/bin/hive --help

```
usage: hive
 -d,--define <key=value>          Variable substitution to apply to hive
                                  commands. e.g. -d A=B or --define A=B
 -e <quoted-query-string>         SQL from command line
 -f <filename>                    SQL from files
 -H,--help                        Print help information
 -h <hostname>                    Connecting to Hive Server on remote host
    --hiveconf <property=value>   Use value for given property
    --hivevar <key=value>         Variable substitution to apply to hive
                                  commands. e.g. --hivevar A=B
 -i <filename>                    Initialization SQL file
 -p <port>                        Connecting to Hive Server on port number
 -S,--silent                      Silent mode in interactive shell
 -v,--verbose                     Verbose mode (echo executed SQL to the
                                  console)
```

CLI—hive外部资源

小象科技 ChinaHadoop.cn

- 外部资源：HQL运行时需要的jar包、二进制文件、文本文件、压缩文件等
- 外部资源需分发到集群的各个节点上使用
- 三种外部资源：
  - ✓ **FILE：普通文件，Hadoop不会进行任何处理**
  - ✓ **JAR：jar包，Hadoop自动将其加入CLASSPATH中**
  - ✓ **ARCHIVE：归档文件，Hadoop可识别“.tgz”、“.tar.gz”、“.zip”等结尾的文件，并自动解压**

CLI—hive外部资源

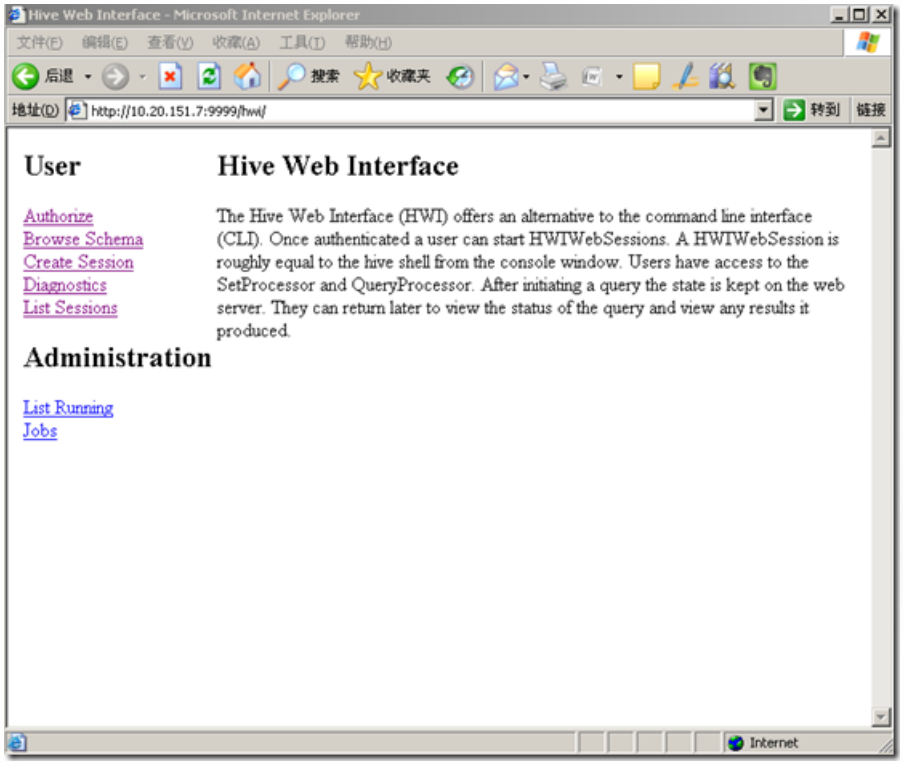
小象科技 ChinaHadoop.cn

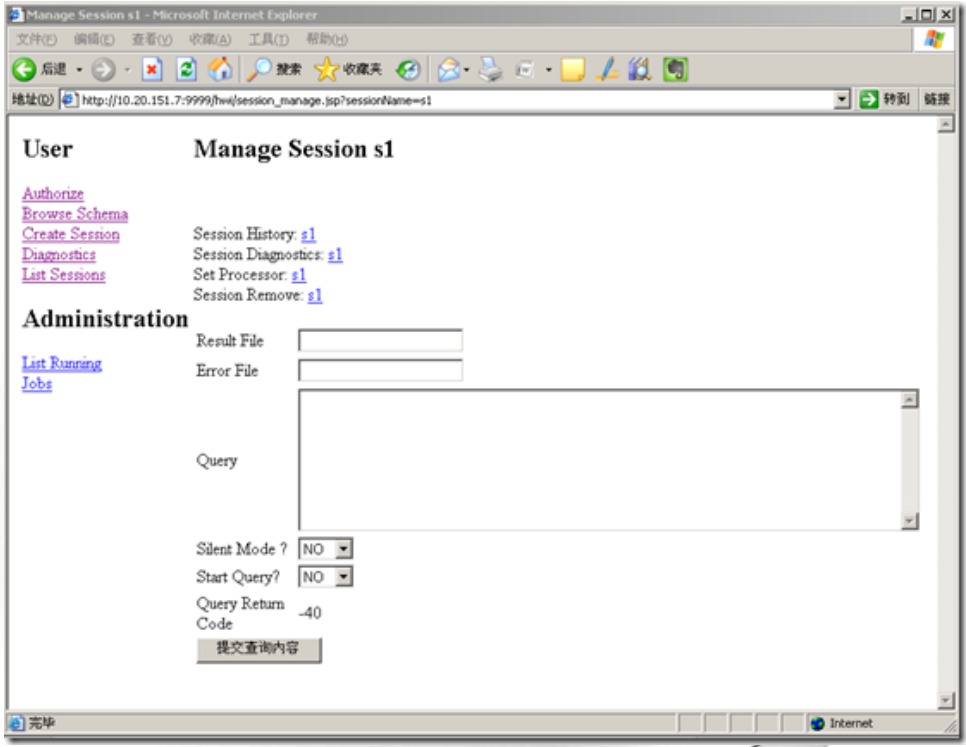
- 三种操作：
- ✓ ADD { FILE[S] | JAR[S] | ARCHIVE[S] } <filepath1> [<filepath2>]*
- ✓ LIST { FILE[S] | JAR[S] | ARCHIVE[S] } [<filepath1> <filepath2> ..]
- ✓ DELETE { FILE[S] | JAR[S] | ARCHIVE[S] } [<filepath1> <filepath2> ..]

```
hive> add FILE /tmp/tt.py;
hive> list FILES;
/tmp/tt.py
hive> select from networks a
            MAP a.networkid
            USING 'python tt.py' as nn where a.ds = '2009-01-04' limit 10;
```

Hive Web UI

小象科技 ChinaHadoop.cn





Hive客户端程序

小象科技 ChinaHadoop.cn

- 方法1：提供JDBC/ODBC访问方式
- 方法2：采用开源软件Thrift实现C/S模型
  - ✓ 支持任何语言编写客户端程序

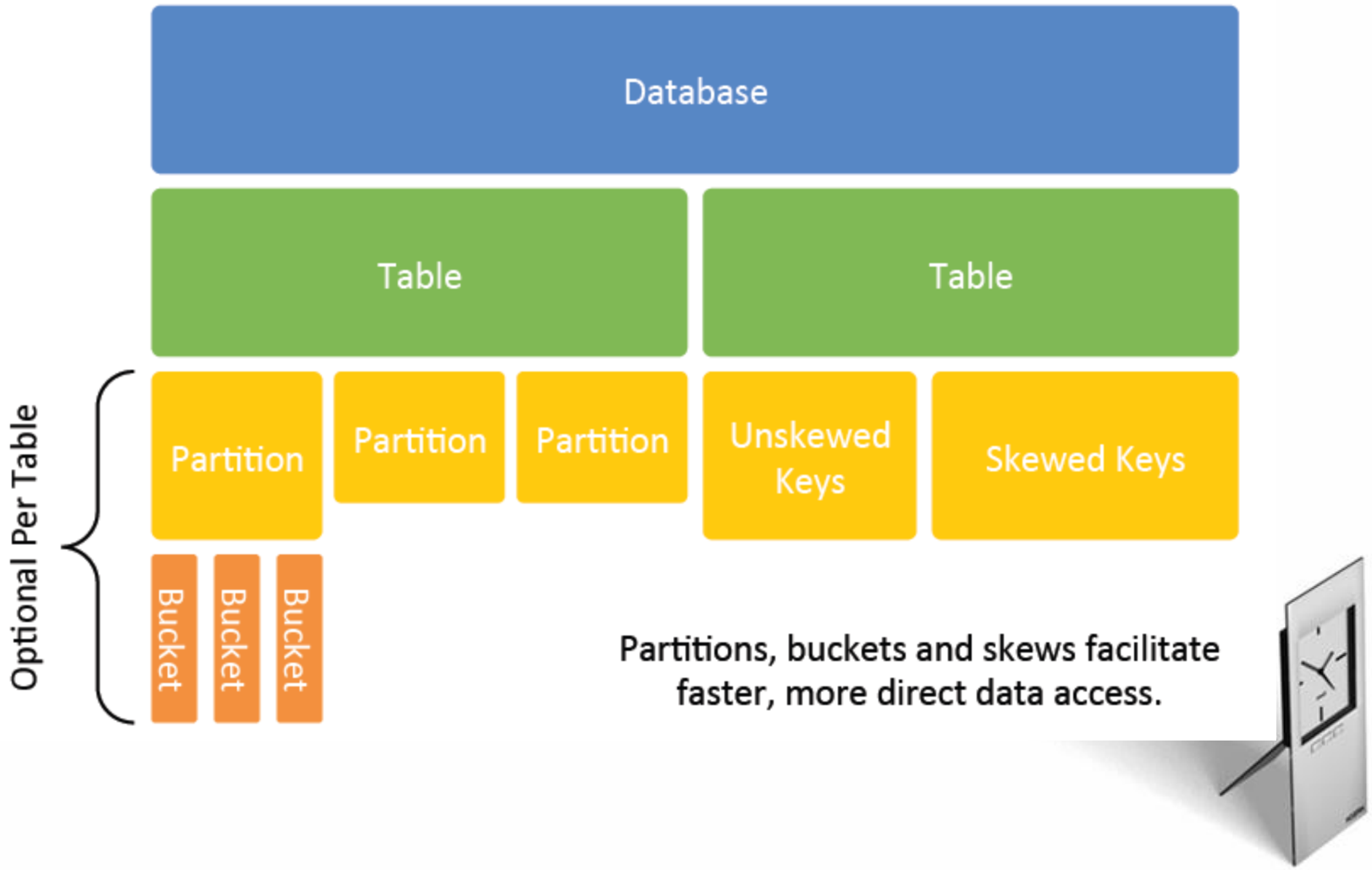
```php
<?php
// set THRIFT_ROOT to php directory of the hive distribution
$GLOBALS['THRIFT_ROOT'] = '/lib/php/';
// load the required files for connecting to Hive
require_once $GLOBALS['THRIFT_ROOT'] . 'packages/hive_service/ThriftHive.php';
require_once $GLOBALS['THRIFT_ROOT'] . 'transport/TSocket.php';
require_once $GLOBALS['THRIFT_ROOT'] . 'protocol/TBinaryProtocol.php';
// Set up the transport/protocol/client
$transport = new TSocket('localhost', 10000);
$protocol = new TBinaryProtocol($transport);
$client = new ThriftHiveClient($protocol);
$transport->open();

// run queries, metadata calls etc
$client->execute('SELECT * from src');
var_dump($client->fetchAll());
$transport->close();
```

主要内容

小象科技 ChinaHadoop.cn

1. Hive背景及应用场景
2. Hive基本架构
3. Hive使用方式
4. HQL查询语句
5. Hive总结及其类似开源系统

数据模型


Database
Table
Table
Partition
Partition
Partition
Unskewed Keys
Skewed Keys
Bucket
Bucket
Bucket
Optional Per Table
Partitions, buckets and skews facilitate faster, more direct data access.

数据类型（不断增加中……）

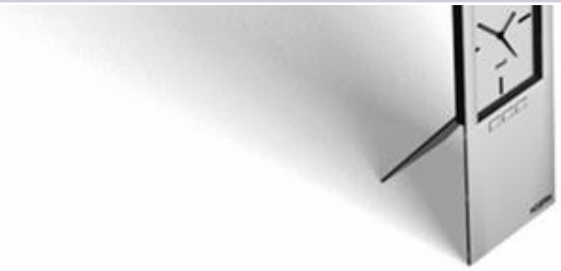
小象科技 ChinaHadoop.cn

类型	大小	举例
TINYINT	1 byte 有符号整型	20
SMALLINT	2 byte 有符号整型	20
INT	4 byte 有符号整型	20
BIGINT	8 byte 有符号整型	20
BOOLEAN	布尔类型	true
FLOAT	单精度浮点型	3.14159
DOUBLE	双度浮点型	3.14159
STRING(CHAR, VARCHAR)	字符串	‘Now is the time’, “for all”
TIMESTAMP（Date）	整型、浮点型或字符串	1327882394 (Unix epoch seconds), 1327882394.123456789 (Unix epoch seconds plus nanoseconds),
BINARY	字节数组	

数据类型

小象科技 ChinaHadoop.cn

类型	解释	举例
STRUCT	与C/C++中的结构体类似，可通过“.”访问每个域的值，比如STRUCT {first STRING; last STRING}，可通过name.first访问第一个成员。	struct('John', 'Doe')
MAP	存储key/value对，可通过[‘key’]获取每个key的值，比如‘first’→‘John’ and ‘last’→‘Doe’,可通过name[‘last’]获取last name。	map('first', 'John', 'last', 'Doe')
ARRAY	同种类型的数据集合，从0开始索引，比如[‘John’, ‘Doe’],，可通过name[1]获取“Doe”。	array('John', 'Doe')



数据定义语句（DDL）

小象科技 ChinaHadoop.cn

- Create/Drop/Alter Database
- Create/Drop/Truncate Table
- Alter Table/Partition/Column
- Create/Drop/Alter View
- Create/Drop/Alter Index
- Create/Drop Function
- Create/Drop/Grant/Revoke Roles and Privileges
- Show
- Describe

数据定义语句（DDL）

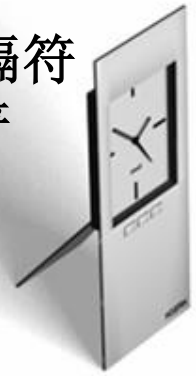
CREATE [EXTERNAL] **TABLE** [IF NOT EXISTS] table_name
(col_name data_type, ...)
[**PARTITIONED BY** (col_name data_type, ...)]
[**CLUSTERED BY** (col_name, col_name, ...) [**SORTED BY**
(col_name [ASC|DESC], ...)] INTO num_buckets BUCKETS]
[**SKEWED BY** (col_name, col_name, ...)]
[[**ROW FORMAT** row_format] [**STORED AS** file_format]]
[**LOCATION** hdfs_path]

数据定义语句（DDL）

小象科技 ChinaHadoop.cn

```
CREATE TABLE employees (
  name         STRING,
  salary       FLOAT,
  subordinates ARRAY<STRING>,
  deductions   MAP<STRING, FLOAT>,
  address      STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\001'
COLLECTION ITEMS TERMINATED BY '\002'
MAP KEYS TERMINATED BY '\003'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

- **ROW FORMAT DELIMITED：保留关键字**
- **FIELDS TERMINATED BY：列分隔符**
- **COLLECTION ITEMS TERMINATED BY ：元素间分隔符**
- **MAP KEYS TERMINATED BY：key/value对间的分隔符**
- **LINES TERMINATED BY：行分隔符**



数据定义语句（DDL）

小象科技 ChinaHadoop.cn

分隔符	解释
\n	记录间的分割符，默认一行一条记录
^A (“control” A)	列分隔符，通常写成“\001”
^B	ARRAY或STRUCT中元素分隔符，或MAP中key与value分隔符，通常写成“\002”
^C	MAP中key/value对间的分隔符，通常写成“\003”

John Doe^A100000.0^AMary Smith^BTodd Jones^AFederal Taxes^C.2^BState Taxes^C.05^BInsurance^C.1^A1 Michigan Ave.^BChicago^BIL^B60600

Mary Smith^A80000.0^ABill King^AFederal Taxes^C.2^BState Taxes^C.05^BInsurance^C.1^A100 Ontario St.^BChicago^BIL^B60601

Todd Jones^A70000.0^AFederal Taxes^C.15^BState Taxes^C.03^BInsurance^C.1^A200 Chicago Ave.^BOak Park^BIL^B60700

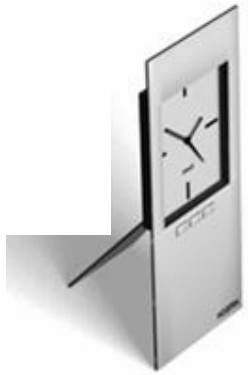
Bill King^A60000.0^AFederal Taxes^C.15^BState Taxes^C.03^BInsurance^C.1^A300 Obscure Dr.^BObscuria^BIL^B60100

Hive Partition与Bucket

小象科技 ChinaHadoop.cn

- Partition
  - ✓ 为减少不必要的暴力数据扫描，可以对表进行分区
  - ✓ 为避免产生过多小文件，建议只对离散字段进行分区

```
CREATE TABLE employees (
  name         STRING,
  salary       FLOAT,
  subordinates ARRAY<STRING>,
  deductions   MAP<STRING, FLOAT>,
  address      STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
)
PARTITIONED BY (country STRING, state STRING);
...
.../employees/country=CA/state=AB
.../employees/country=CA/state=BC
...
.../employees/country=US/state=AL
.../employees/country=US/state=AK
...

SELECT * FROM employees
WHERE country = 'US' AND state = 'IL';
```



Hive Partition与Bucket

小象科技 ChinaHadoop.cn

- Bucket
  - ✓对于值较多的字段，可将其分成若干个Bucket
  - ✓可结合Partition与Bucket使用

```
CREATE TABLE weblog (user_id INT, url STRING, source_ip STRING)
PARTITIONED BY (dt STRING)
CLUSTERED BY (user_id) INTO 96 BUCKETS;

SET hive.enforce.bucketing = true;

FROM raw_logs
INSERT OVERWRITE TABLE weblog
PARTITION (dt='2009-02-25')
SELECT user_id, url, source_ip WHERE dt='2009-02-25';
```

Hive数据格式

小象科技 ChinaHadoop.cn

- 由用户自定义
  - ✓默认是文本文件（TEXTFILE）
  - ✓文本文件，用户需显示指定分隔符
- 其他已支持的格式
  - ✓SequenceFile
  - ✓Avro
  - ✓ORC/Parquet
  - ✓用户自定义(InputFormat与OutputFormat)
- 支持数据压缩
  - ✓Bzip、Gzip
  - ✓LZO
  - ✓Snappy

应用举例1：日志处理

小象科技 ChinaHadoop.cn

- 某网站日志格式如下：

```
10180,2009-03-01 00:34:44,20090301,1001,12911621
10180,2009-03-01 00:34:46,20090301,0,841842809
10180,2009-03-01 00:34:48,20090301,10180,22395900
10180,2009-03-01 00:34:50,20090301,10180,867110912
```

- 表定义如下：

```
CREATE TABLE logs(
   domain_id INT ,
   log_time STRING ,
   log_date STRING,
   log_type INT,
   uin BIGINT )
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY ',‘
   LOCATION '/user/hivetest/logs';
```

应用举例1：日志处理

小象科技 ChinaHadoop.cn

- 将数据拷贝到数据库目录下：

hadoop fs -put ./20130301.small /user/hivetest/logs/small1
hadoop fs -put ./20130302.small /user/hivetest/logs/small2

- 执行HQL语句：

select * from logs ;

数据操作语句（DML）

小象科技 ChinaHadoop.cn

- 数据加载与插入语句
  - ✓ LOAD
  - ✓ INSERT
- 数据查询语句
  - ✓ SELECT
- 查看HQL执行计划
  - ✓ explain
- 表/分区导入导出
  - ✓ Export/Import

数据加载与插入语句

小象科技 ChinaHadoop.cn

- **Load data**
  - ✓ **LOAD DATA** [LOCAL] **INPATH** 'filepath' [OVERWRITE] **INTO TABLE** tablename[PARTITION (partcol1=val1, partcol2=val2 ...)]
- **Insert**
  - ✓ **INSERT OVERWRITE TABLE** tablename[PARTITION (partcol1=val1, partcol2=val2 ...)] select_statement **FROM** from_statement
- **Multiple insert**
  - ✓ **FROM** from_statement

  INSERT OVERWRITE TABLE tablename1 [PARTITION...)] select_statement1

  [INSERT OVERWRITE TABLE tablename2 [PARTITION ...] select_statement2] ...

数据加载与插入语句

小象科技 ChinaHadoop.cn

- **Load data**
  - ✓ 当数据被加载至表中时，不会对数据进行任何转换。Load 操作只是将数据复制/移动至 Hive 表对应的位置。
  - ✓ 默认每个表一个目录，比如数据库dbtest中，表名为tbtest，则数据存放位置为：${metastore.warehouse.dir}/dbtest.db/tbtest
  - ✓ metastore.warehouse.dir默认值是/user/hive/warehouse

几个实例

小象科技 ChinaHadoop.cn

```
LOAD DATA LOCAL INPATH '${env:HOME}/california-employees'
OVERWRITE INTO TABLE employees
PARTITION (country = 'US', state = 'CA');
```

```
INSERT OVERWRITE TABLE employees
PARTITION (country = 'US', state = 'OR')
SELECT * FROM staged_employees se
WHERE se.cnty = 'US' AND se.st = 'OR';
```

```
FROM staged_employees se
INSERT OVERWRITE TABLE employees
  PARTITION (country = 'US', state = 'OR')
  SELECT * WHERE se.cnty = 'US' AND se.st = 'OR'
INSERT OVERWRITE TABLE employees
  PARTITION (country = 'US', state = 'CA')
  SELECT * WHERE se.cnty = 'US' AND se.st = 'CA'
INSERT OVERWRITE TABLE employees
  PARTITION (country = 'US', state = 'IL')
  SELECT * WHERE se.cnty = 'US' AND se.st = 'IL';
```

数据查询语句SELECT

小象科技 ChinaHadoop.cn

SELECT [ALL | DISTINCT] select_expr,
select_expr, ...
FROM table_reference
[**WHERE** where_condition]
[**GROUP BY** col_list]
[**CLUSTER BY** col_list| [**DISTRIBUTE BY** col_list] [**SORT BY** col_list] | [**ORDER BY** col_list]]
[**LIMIT** number]

- **不支持having和exist in操作, 可转换为LEFT SEMI JOIN操作**

数据查询语句SELECT

小象科技 ChinaHadoop.cn

- **Join（仅支持等值连接）**
  - ✓ INNER JOIN
  - ✓ LEFT OUTER JOIN
  - ✓ RIGHT OUTER JOIN
  - ✓ FULL OUTER JOIN
  - ✓ LEFT SEMI-JOIN
  - ✓ Map-side Joins
- **不支持非等值的连接**

两种分布式Join算法

小象科技 ChinaHadoop.cn

- **Map-side Join（Broadcast join）**
  - ✓ **Join操作在map task中完成，因此无需启动reduce task；**
  - ✓ **适合一个大表，一个小表的连接操作**
  - ✓ **思想：小表复制到各个节点上，并加载到内存中；大表分片，与小表完成连接操作**
- **Reduce-side Join（shuffle join）**
  - ✓ **Join操作在reduce task中完成；**
  - ✓ **适合两个大表连接操作**
  - ✓ **思想：map端按照连接字段进行hash，reduce 端完成连接操作**

HQL中两种特殊Join

小象科技 ChinaHadoop.cn

- **Map-side Join（Broadcast join）**

SELECT /*+ MAPJOIN(b) */ a.key, a.value

FROM a join b on a.key = b.key

- **LEFT SEMI JOIN（左半连接）**

```
SELECT a.key, a.value
FROM a
WHERE a.key in
 (SELECT b.key
  FROM B);
```

```
SELECT a.key, a.val
FROM a LEFT SEMI JOIN b on (a.key = b.key)
```

Order By与Sort By

- **Order by**
  - ✓ 启动一个reduce task
  - ✓ 数据全局有序
  - ✓ 速度可能会非常慢
  - ✓ Strict模式下，必须与limit连用
- **Sort by**
  - ✓ 可以有多个reduce task
  - ✓ 每个Reduce Task内部数据有序，但全局无序
  - ✓ 通常与distribute by

Distribute by与Cluster by

小象科技 ChinaHadoop.cn

- **distribute by**
  - ✓ 相当于MapReduce中的paritioner，默认是基于hash实现的；
  - ✓ 与sort by连用，可发挥很好的作用
  - ✓ 举例：

```
hive> SELECT s.ymd, s.symbol, s.price_close
    > FROM stocks s
    > DISTRIBUTE BY s.symbol
    > SORT BY  s.symbol ASC, s.ymd ASC;
```

- **cluster by**
  - ✓ 当distribute by与sort by（降序）连用，且跟随的字段相同时，可使用cluster by简写；
  - ✓ 举例：

```
hive> SELECT s.ymd, s.symbol, s.price_close
    > FROM stocks s
    > CLUSTER BY s.symbol;
```

Transform语法（Streaming）

小象科技 ChinaHadoop.cn

- **Hive允许使用任意语言编写Mapper和Reducer，并嵌到HQL中使用**
- **实现机制类似于Hadoop Streaming，在Java进程中额外启动一个线程运行脚本/二进制程序，并通过标准输入输出进行数据传递**
- **使用操作符TRANSFORM**

Transform语法（Streaming）

小象科技 ChinaHadoop.cn

- **解析一定key/value格式的数据，并对其格式化**

```
k1=v1,k2=v2
k4=v4,k5=v5,k6=v6
k7=v7,k7=v7,k3=v7
```

```
k1	v1
k2	v2
k4	v4
k5	v5
k6	v6
k7	v7
k7	v7
k3	v7
```

- **编写一下perl脚本：**

```
#!/usr/bin/perl
while (<STDIN>) {
    my $line = $_;
    chomp($line);
    my @kvs = split(/,/, $line);
    foreach my $p (@kvs) {
        my @kv = split(/=/, $p);
        print $kv[0] . "\t" . $kv[1] . "\n";
    }
}
```

- **使用该脚本：**

```
hive> ADD FILE /home/dongxicheng/split_kv.pl;
hive> SELECT TRANSFORM (line)
> USING 'perl split_kv.pl'
> AS (key, value) FROM kv_data;
```

用户自定义函数（UDF）

小象科技 ChinaHadoop.cn

- **UDF：扩展HQL能力的一种方式**
- **三种UDF：**
  - ✓ **普通UDF（1对1）**
  - ✓ **UDAF：用户自定义聚集函数（多对1）**
  - ✓ **UDTF：用户自定义产生表函数（1对多）**
- **函数相关操作：**
  - ✓ **SHOW FUNCTIONS;**
  - ✓ **DESCRIBE FUNCTION concat;**
  - ✓ **DESCRIBE FUNCTION EXTENDED concat;**
  - ✓ **SELECT concat(column1,column2) AS x FROM table;**

用户自定义函数（UDF）

小象科技 ChinaHadoop.cn

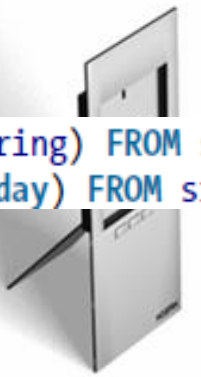
```java
public class UDFZodiacSign extends UDF{

  private SimpleDateFormat df;

  public UDFZodiacSign(){
    df = new SimpleDateFormat("MM-dd-yyyy");
  }

  public String evaluate( Date bday ){
    return this.evaluate( bday.getMonth(), bday.getDay() );
  }

  public String evaluate(String bday){
    Date date = null;
    try {
      date = df.parse(bday);
    } catch (Exception ex) {
      return null;
    }
    return this.evaluate( date.getMonth()+1, date.getDay() );
  }
```

```java
  public String evaluate( Integer month, Integer day ){
    if (month==1) {
      if (day < 20 ){
        return "Capricorn";
      } else {
        return "Aquarius";
      }
    }
    if (month==2){
      if (day < 19 ){
        return "Aquarius";
      } else {
        return "Pisces";
      }
    }
    /* ...other months here */
    return null;
  }
}
```

```
hive> ADD JAR /full/path/to/zodiac.jar;
hive> CREATE TEMPORARY FUNCTION zodiac
    > AS 'org.apache.hadoop.hive.contrib.udf.example.UDFZodiacSign';
```

```
SELECT zodiac(date_string) FROM src;
SELECT zodiac(month, day) FROM src;
```

```
hive> DROP TEMPORARY FUNCTION IF EXISTS zodiac;
```



用户自定义函数（UDF）

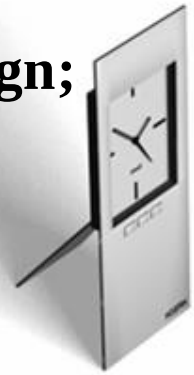
小象科技 ChinaHadoop.cn

- **将函数永久加入 HQL中：**
  - ✓ 修改Hive的函数注册代码
  - ✓ 重新编译hive代码
  - ✓ 重新部署
- **修改hive注册代码：**
  - ✓ 找到以下文件：

ql/src/java/org/apache/hadoop/hive/ql/exec/FunctionRegistry.java

  - ✓ 添加新的注册函数myfunc：

```
org.apache.hadoop.hive.contrib.udf.example.UDFZodiacSign;
...
registerUDF("parse_url", UDFParseUrl.class, false);
registerUDF("pzodiac", UDFZodiacSign.class, false);
```



用户自定义函数（UDF）

小象科技 ChinaHadoop.cn

- **编写复杂的UDF、UDAF和UDTF：**
  - ✓ **继承类GenericUDF**
  - ✓ **实现initialize、evaluate等函数**
- **相关示例：**
  - ✓ **UDAF：**
    http://svn.apache.org/repos/asf/hive/branches/branch-0.13/ql/src/java/org/apache/hadoop/hive/ql/udf/generic/GenericUDAFAverage.java
  - ✓ **UDTF：**
    http://svn.apache.org/repos/asf/hive/branches/branch-0.13/ql/src/java/org/apache/hadoop/hive/ql/udf/generic/GenericUDTFExplode.java

其他问题

- HQL支持索引吗？
  - ✓ HQL执行过程主要是并行地暴力扫描。目前Hive仅支持单表索引，但提供了索引创建接口和调用方法，可由用户根据需要实现索引结构；
- HQL支持update操作吗？
  - ✓ 不支持。Hive底层是HDFS，HDFS仅支持追加操作，不支持随机写；
- Skew Data处理机制？
  - ✓ 指定skew 列：CREATE TABLE list_bucket_single (key STRING, value STRING) SKEWED BY (key) ON (1,5,6);
  - ✓ 为skew task分配更多资源（TODO）
  - ✓ 将skew task分解成多个task，再合并结果（TODO）

Hive On HBase

- 使用HQL处理HBase中的数据
  - ✓ 比直接通过HBase API存取数据方便；
  - ✓ 但性能更低，相当于把在线处理转为批处理
- 存在问题
  - ✓ 不够成熟；
  - ✓ 不能按时间戳获取数据，默认总是取最新的数据

Hive On HBase实例

小象科技 ChinaHadoop.cn

```
CREATE TABLE hbase_table_1(key int, value1 string, value2 int, value3 int)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
"hbase.columns.mapping" = ":key,a:b,a:c,d:e"
);
```

```
CREATE TABLE hbase_table_1(value map<string,int>, row_key int)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
"hbase.columns.mapping" = "cf:,:key"
);
```

主要内容

小象科技 ChinaHadoop.cn

1. Hive背景及应用场景
2. Hive基本架构
3. Hive使用方式
4. HQL查询语句
5. Hive总结及其类似开源系统

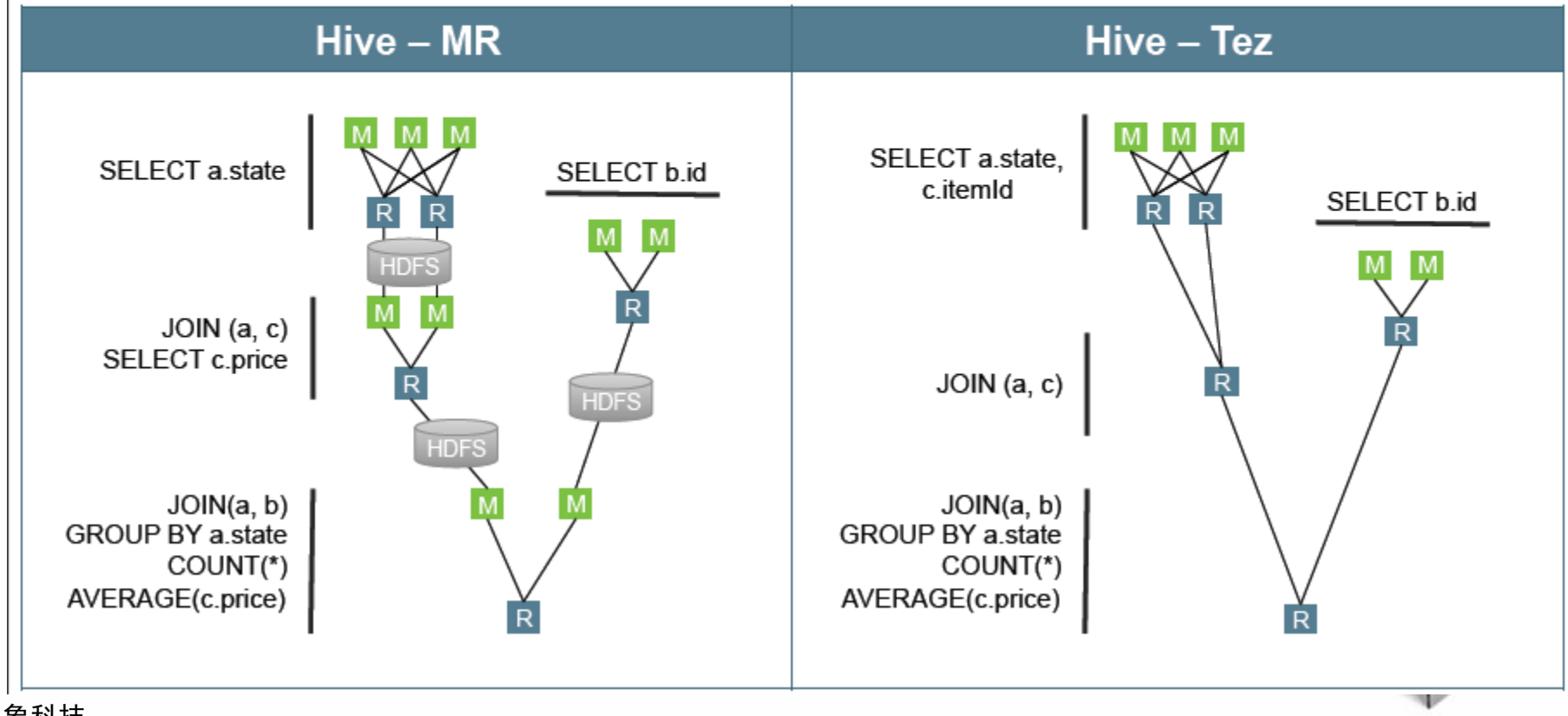
Stinger

- **Tez是一个DAG计算框架，在MapReduce基础上发展起来的，目前是Apache顶级项目；**
- **下一代Hive被称为“Stinger”，其底层的计算引擎将由Tez替换MapReduce；**
- **Tez相比于MapReduce具有众多优势：**
  - ✓ **提供了多种算子（比如Map、Shuffle等）供用户使用；**
  - ✓ **将多个作业合并成一个作业，减少磁盘读写IO；**
  - ✓ **充分利用内存资源。**
- **官方首页：http://tez.incubator.apache.org/**

Hive-MR vs Hive-Tez

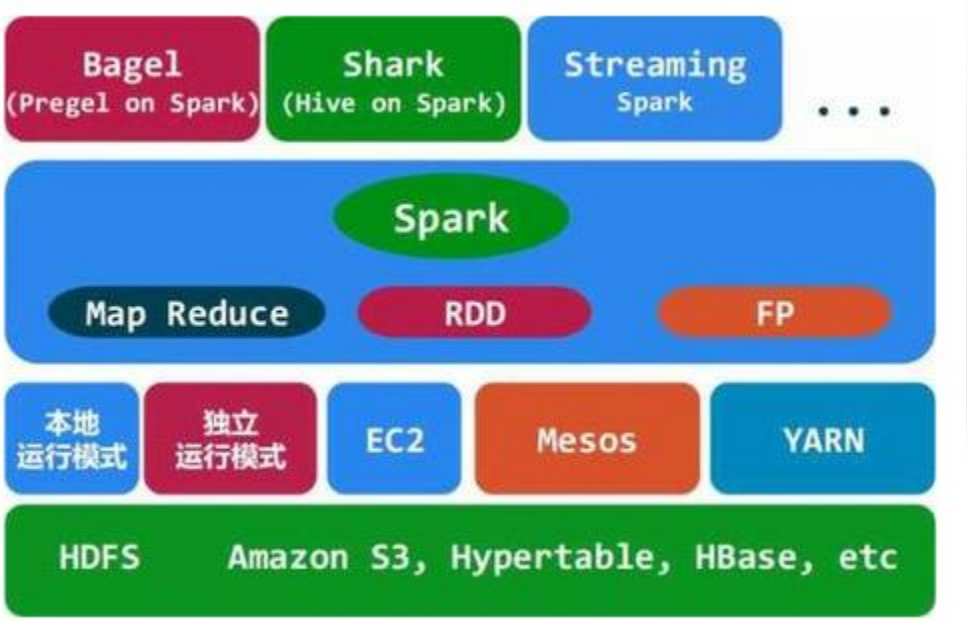

```
SELECT a.state, COUNT(*), AVERAGE(c.price)
FROM a
JOIN b ON (a.id = b.id)
JOIN c ON (a.itemId = c.itemId)
GROUP BY a.state
```

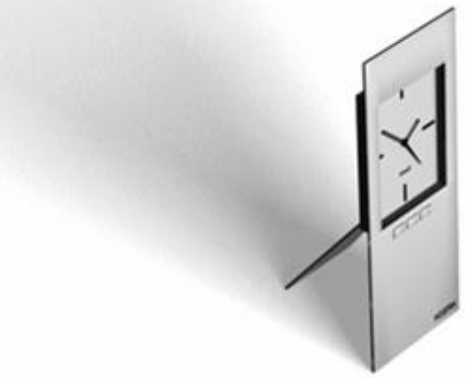
Tez avoids unneeded writes to HDFS



Shark

- **Hive On Spark（http://spark.incubator.apache.org/）；**
- **Spark是一个内存计算框架，相比于MapReduce，效率更加高效（部分测试表明，速度快100x）；**
- **Shark完全兼容Hive，底层计算引擎采用Spark。**



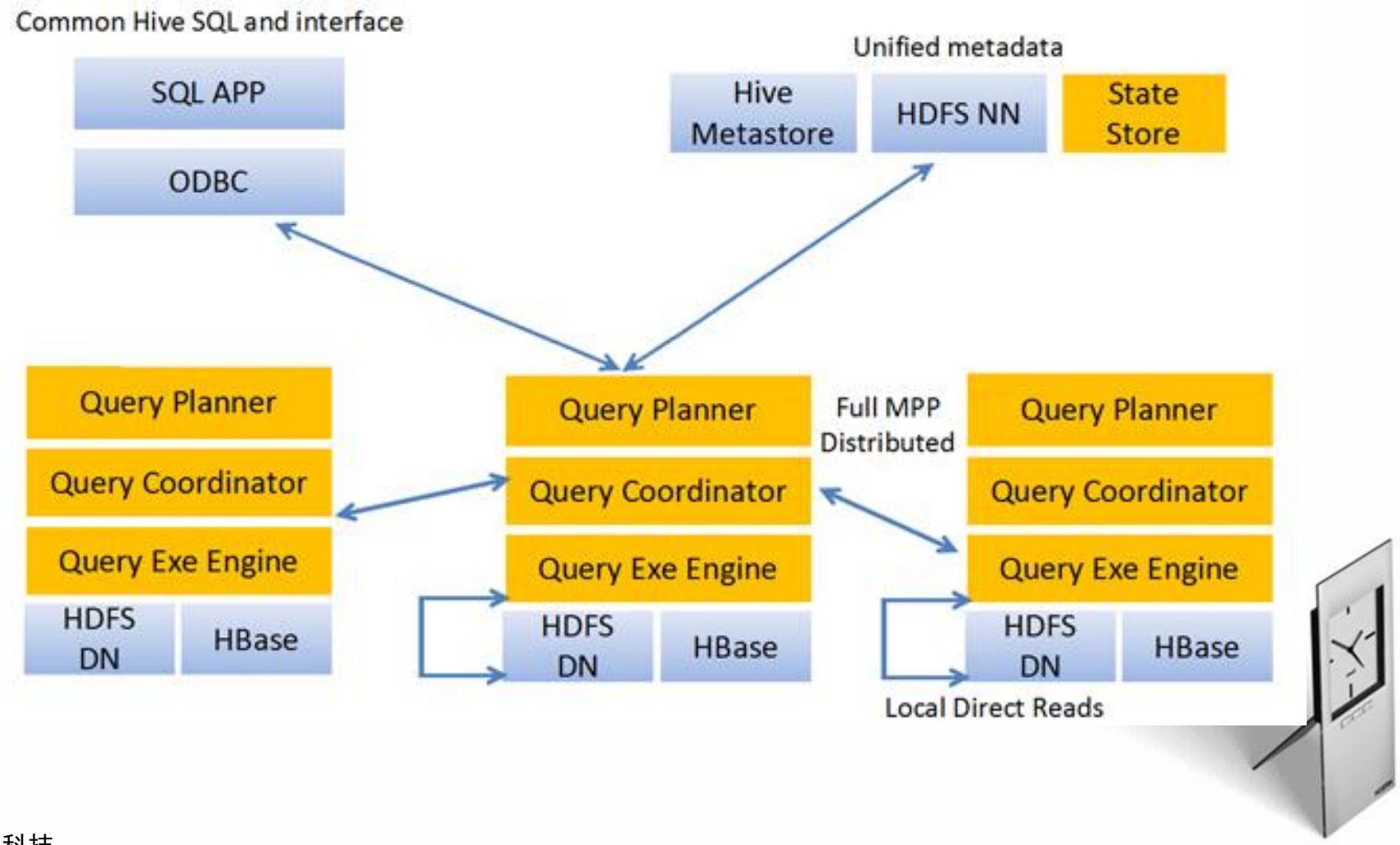


Impala

小象科技 ChinaHadoop.cn

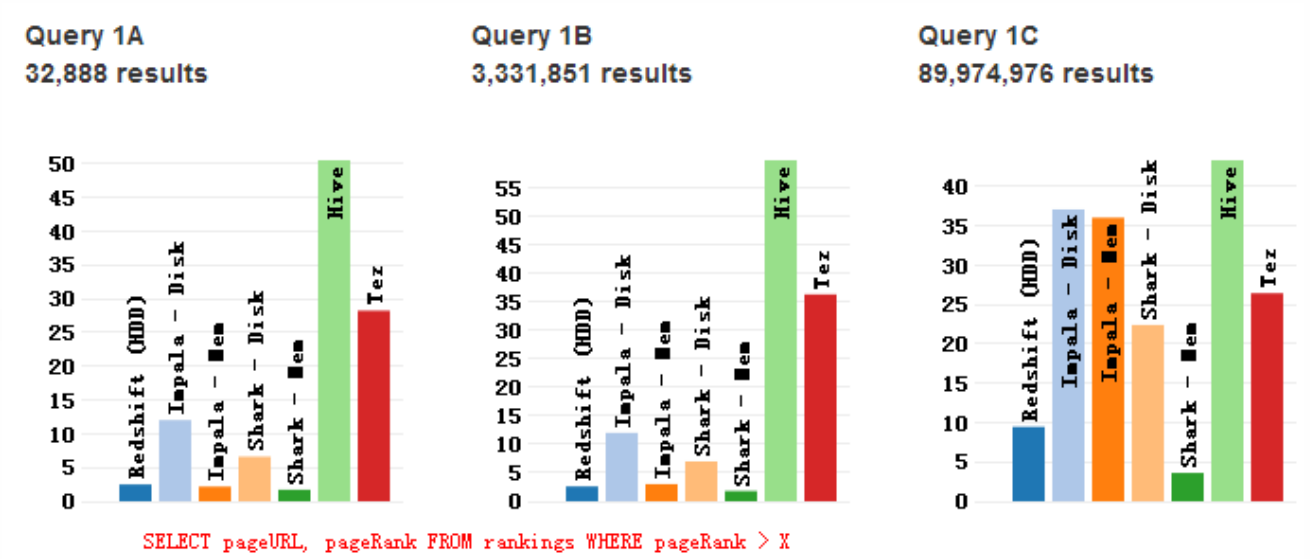
- 底层计算引擎不再采用**MR**，而是使用与**商用并行关系数据库类似的分布式查询引擎；**
- **Impala**可直接处理存储在**HDFS**上的数据，并将结果集再次写入**HDFS；**
- 具有良好的扩展性和容错性；
- 适合快速交互式查询

Impala


Common Hive SQL and interface
SQL APP
ODBC
Unified metadata
Hive Metastore
HDFS NN
State Store
Query Planner
Query Coordinator
Query Exe Engine
HDFS DN
HBase
Query Planner
Query Coordinator
Query Exe Engine
HDFS DN
HBase
Full MPP Distributed
Query Planner
Query Coordinator
Query Exe Engine
HDFS DN
HBase
Local Direct Reads

性能比较

https://amplab.cs.berkeley.edu/benchmark/

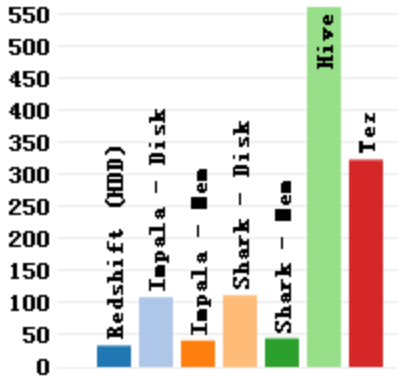


性能比较

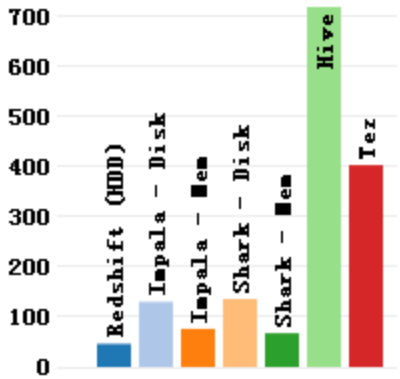
https://amplab.cs.berkeley.edu/benchmark/

```
SELECT sourceIP, totalRevenue, avgPageRank
FROM
  (SELECT sourceIP,
          AVG(pageRank) as avgPageRank,
          SUM(adRevenue) as totalRevenue
    FROM Rankings AS R, UserVisits AS UV
    WHERE R.pageURL = UV.destURL
       AND UV.visitDate BETWEEN Date(`1980-01-01') AND Date(`X')
    GROUP BY UV.sourceIP)
 ORDER BY totalRevenue DESC LIMIT 1
```

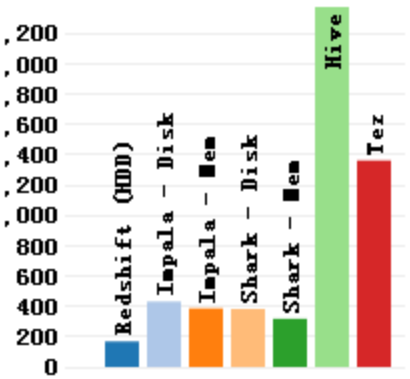
Query 3A
485,312 rows



Query 3B
53,332,015 rows



Query 3C
533,287,121 rows



小象科技 ChinaHadoop.cn

联系我们：

- 新浪微博：ChinaHadoop
- 微信公号：ChinaHadoop

让你的数据产生价值！