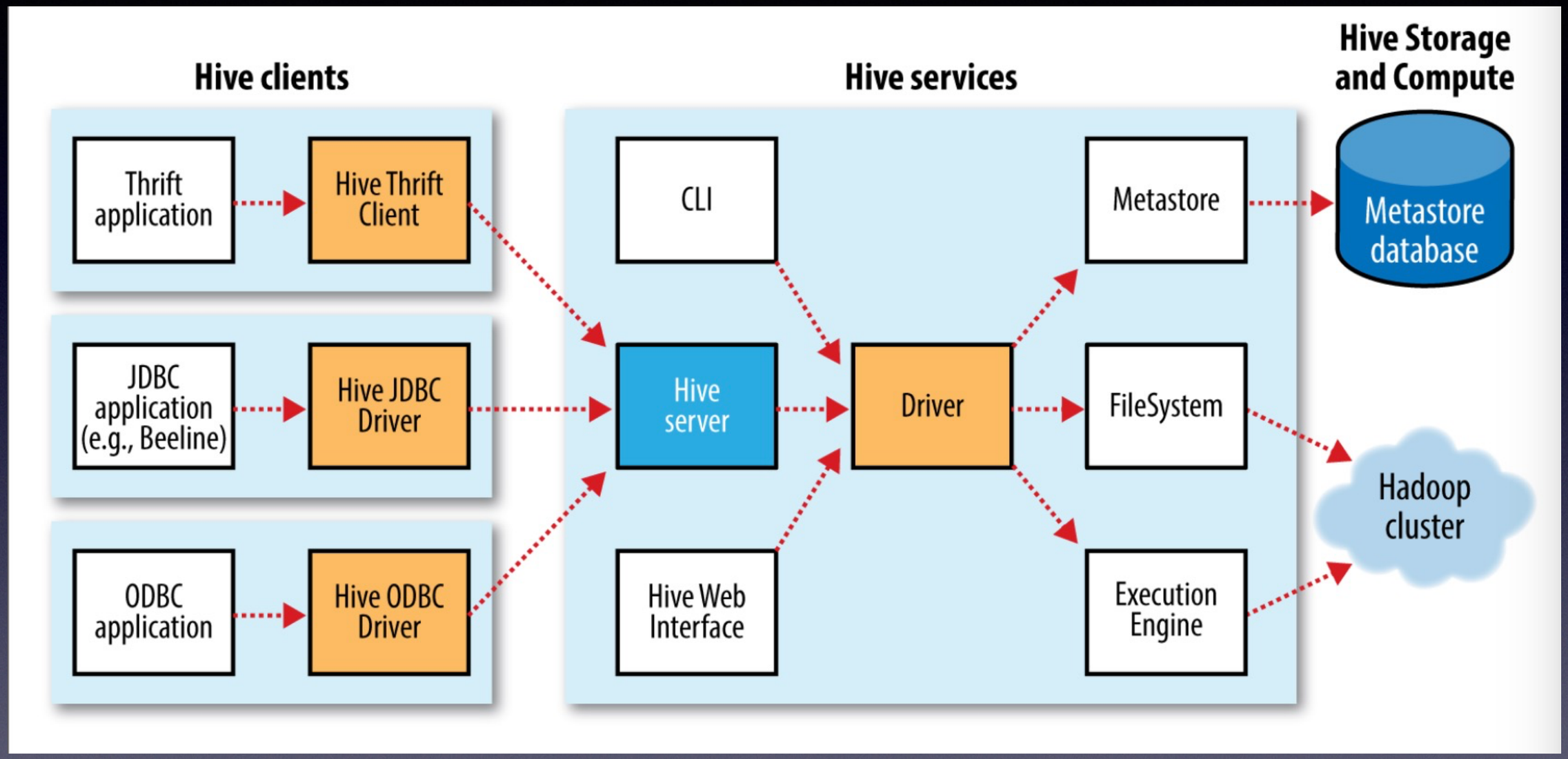
Hive介绍

Hive起源于Facebook，它使得针对Hadoop进行SQL查询成为可能，从而非程序员也可以方便地使用。Hive是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供完整的SQL查询功能，可以将SQL语句转换为MapReduce任务运行。

O'Reilly Media Inc ： Hive是Hadoop生态系统中必不可少的一个工具，它提供了一种SQL（结构化查询语言）方言，可以查询存储在Hadoop分布式文件系统（HDFS）中的数据或其他和Hadoop集成的文件系统，如MapR-FS、Amazon的S3和像HBase（Hadoop数据库）和Cassandra这样的数据库中的数据。

大多数数据仓库应用程序都是使用关系数据库进行实现的，并使用SQL作为查询语言。Hive降低了将这些应用程序转移到Hadoop系统上的难度。凡是会使用SQL语言的开发人员都可以很轻松地学习并使用Hive。如果没有Hive，那么这些用户就必须学习新的语言和工具，然后才能应用到生产环境中。另外，相比其他工具，Hive更便于开发人员将基于SQL的应用程序转移到Hadoop中。如果没有Hive，那么开发者将面临一个艰巨的挑战，如何将他们的SQL应用程序移植到Hadoop上。

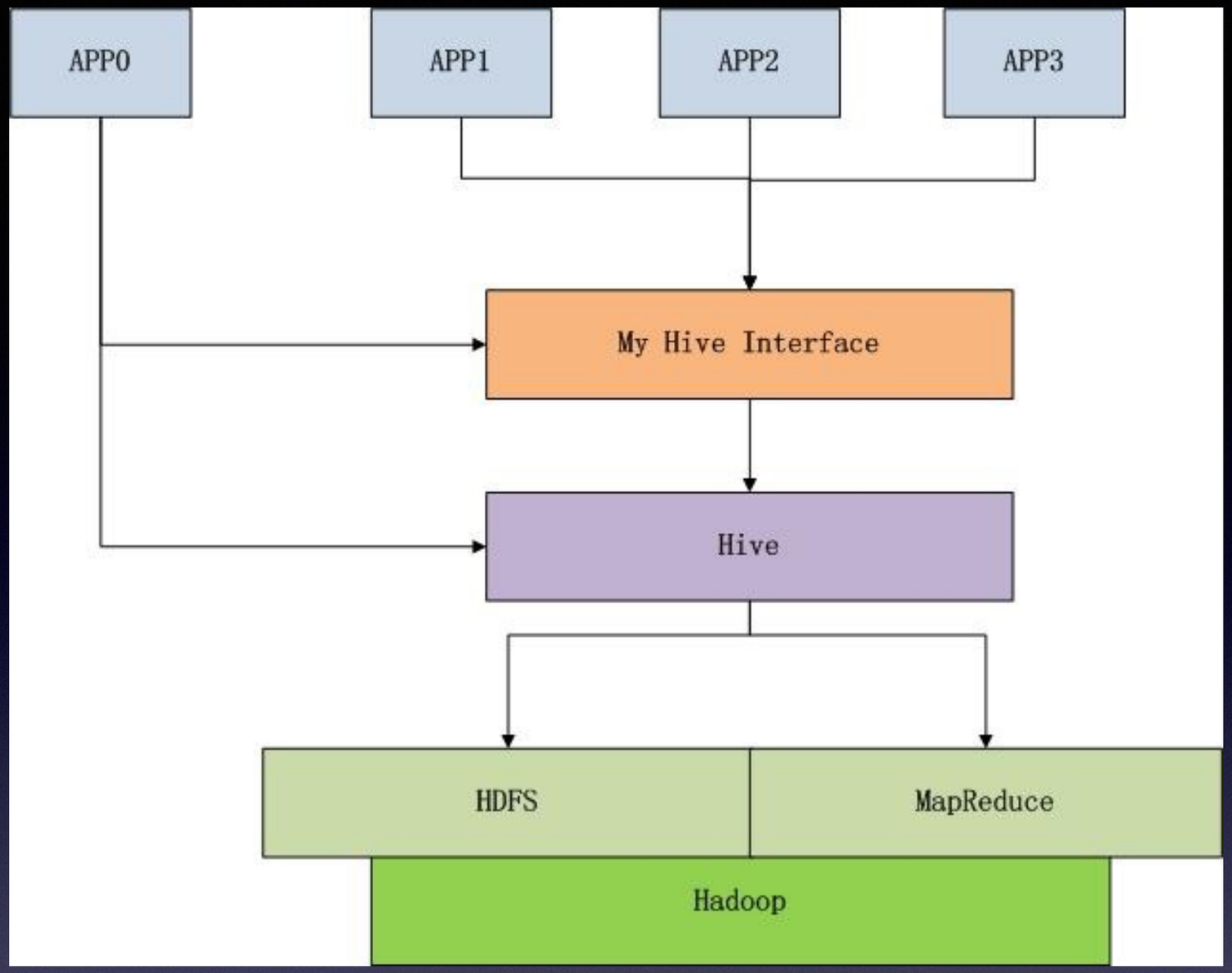

Hive clients
Hive services
Hive Storage and Compute
Thrift application
Hive Thrift Client
JDBC application (e.g., Beeline)
Hive JDBC Driver
ODBC application
Hive ODBC Driver
CLI
Hive server
Hive Web Interface
Driver
Metastore
FileSystem
Execution Engine
Metastore database
Hadoop cluster

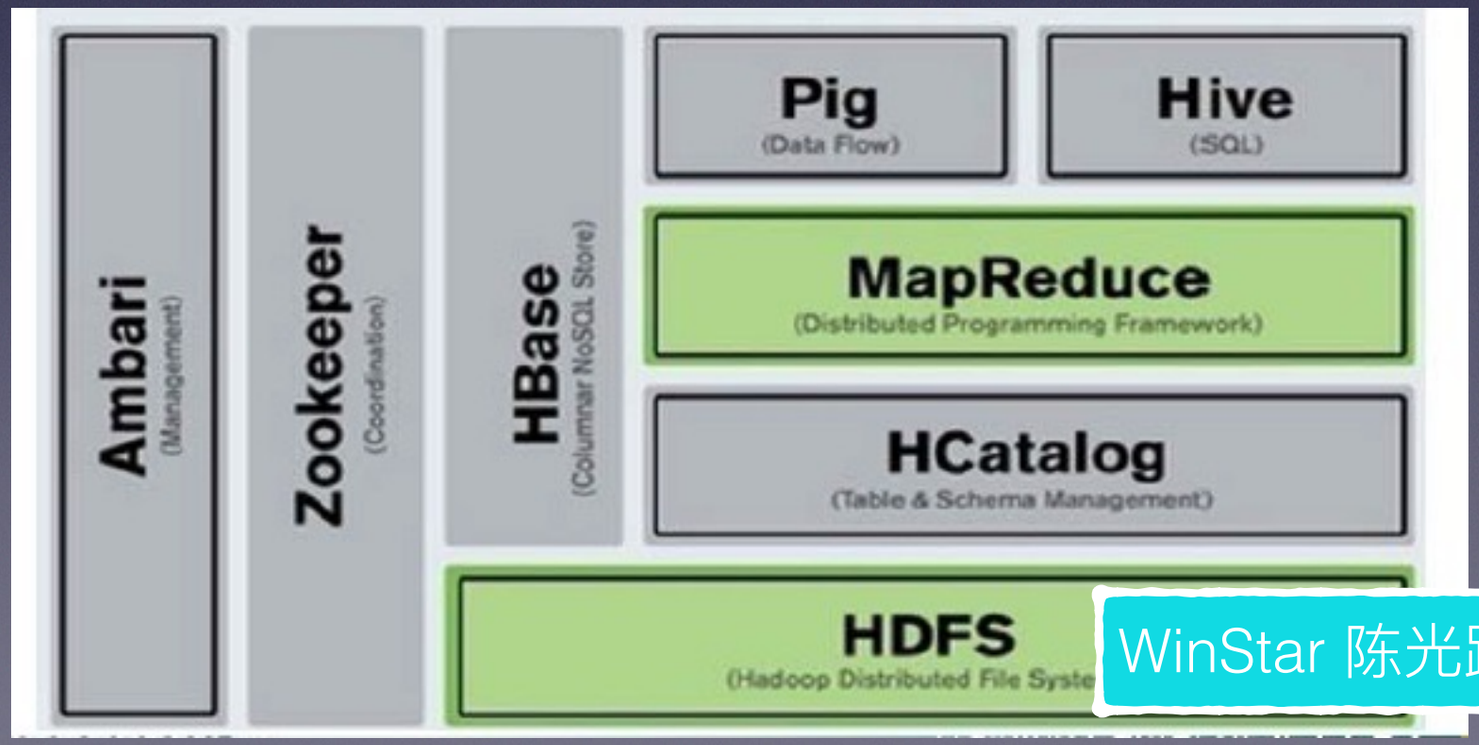
WinStar 陈光跃

- 优点及使用场景

- hive封装了MapReduce的进程，操作接口采用类SQL语法，提供快速开发的能力(简单、容易上手)；
- 在考虑构建Hadoop分析工具时，以Hive作为Hadoop访问接口是最有效的。
- 避免了去写MapReduce，减少开发人员的学习成本；
- 统一的元数据管理，可与impala/spark等共享元数据；
- 易扩展(HDFS+MapReduce：可以扩展集群规模；支持自定义函数)；
- 数据的离线处理；比如：日志分析，海量结构化数据离线分析...
- Hive的执行延迟比较高，因此hive常用于数据分析的，对实时性要求不高的场合；
- Hive优势在于处理大数据，对于处理小数据没有优势，因为Hive的执行延迟比较高。

HIVE在Hadoop家族中的位置





WinStar 陈光跃

本次使用实例

MongoDB、hive与hadoop

MongoDB是后端业务数据库，hadoop平台作为数据平台。将数据从MongoDB导出来，然后传到HDFS，最后用Hive / MR处理

HIVE的安装与环境配置

- 下载地址：http://mirror.bit.edu.cn/apache/hive/ （建议2.3.3，高版本不稳定），之后解压并移入到hadoop目录下。例：mv apache-hive-2.3.3-bin /usr/hadoop/hadoop-3.0.3/hive

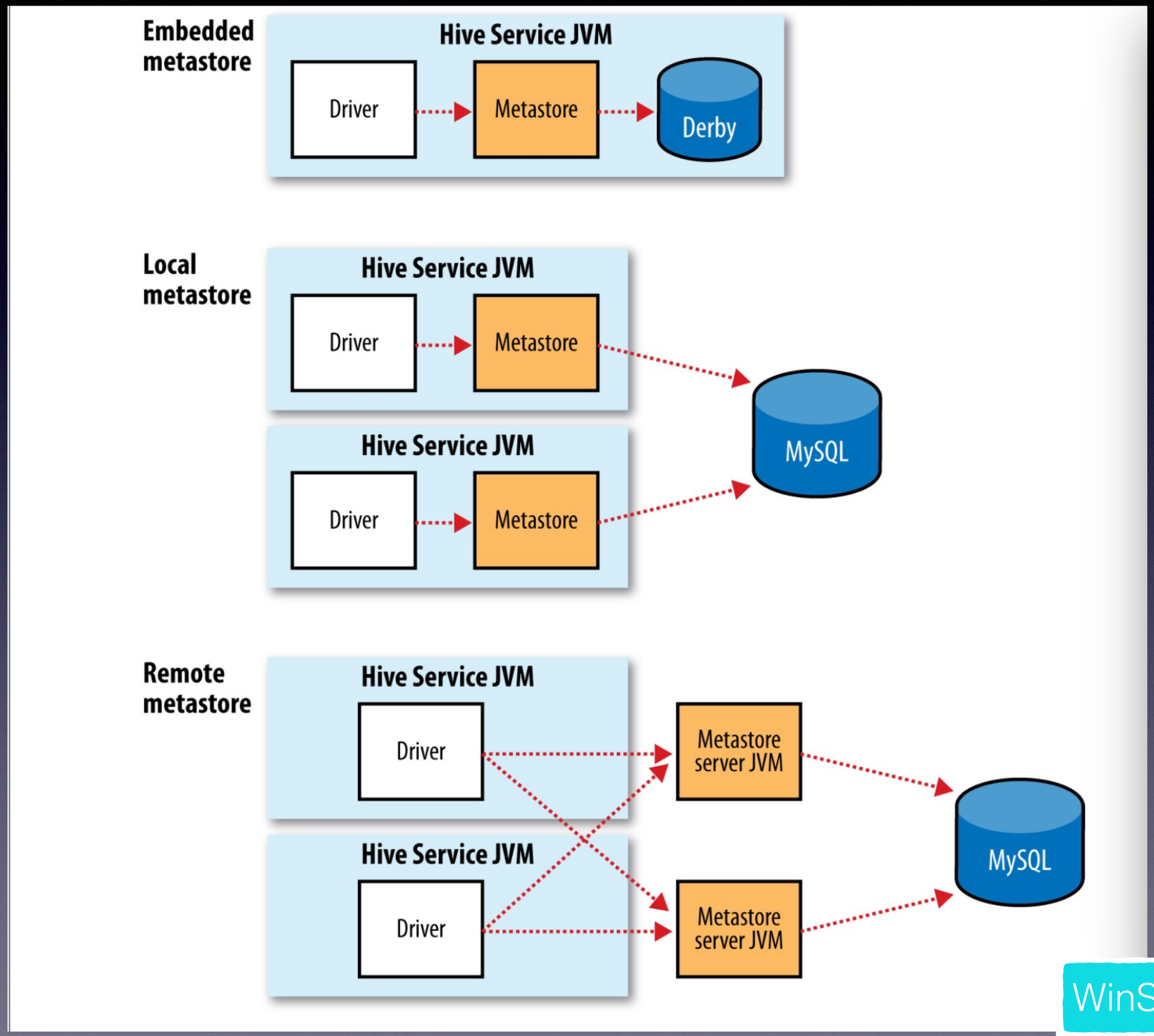
- 修改环境变量：vim /etc/profile

```
#hive_home
export HIVE_HOME=$HADOOP_HOME/hive
export PATH=$PATH:$HIVE_HOME/bin
export CLASSPATH=.:$HIVE_HOME/lib:$CLASSPATH
```

- hive-env.sh： cd $HADOOP_HOME/apache-hive-3.0.0-bin/conf/
  - mv hive-env.sh.template hive-env.sh
  - vim hive-env.sh
  - 找到HADOOP_HOME改为本机的HADOOP _HOME： 例子： HADOOP_HOEM =/usr/hadoop/hadoop-3.0.3

WinStar 陈光跃

Metastore是Hive元数据的中央存储库。



WinStar 陈光跃

- 安装mysql：因为hive默认库时使用内嵌metastore的derby数据库，缺点是一次只能打开一个会话，mysql则可以多用户访问，所以通常选择mysql，以支持多个会话（以及多个用户），此配置称为本地Metastore。
  - apt install mysql-server
  - apt install mysql-client
- 之后在mySQL里面创建用户与元数据表，并赋予所有权限

```
mysql> CREATE USER 'hive' IDENTIFIED BY 'mysql';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'hive'@'%' WITH GRANT
OPTION;
mysql> flush privileges;
mysql> exit;
hadoop@ubuntu:~$ mysql -uhive -pmysql
mysql> create database hive;
```

WinStar 陈光跃

- 修改Hive配置文件：cd $HIVE_HOME/conf
    - mv hive-default.xml.template hive-default.xml
    - mv hive-default.xml.template hive-site.xml
    - vim hive-site.xml

```
<property>
    <name>hive.metastore.local</name>
    <value>true</value>
  </property>
<property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:mysql://localhost/hive?characterEncoding=UTF-8&amp;useSSL=false</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>com.mysql.cj.jdbc.Driver</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionUserName</name>
    <value>hive</value>
  </property>
  <property>
    <name>javax.jdo.option.ConnectionPassword</name>
    <value>mysql</value>
  </property>
```

JDBC版本与MySQL版本不兼容,MySQL的版本更高一些

WinStar 陈光跃

配置完成

- 命令行输入：hive进入hive环境。
- 若是存在slf4j-jar包冲突（hive与hadoop均自带），删除一个即可。

WinStar 陈光跃

从MongoDB导出数据

- 有三种方法：
  - 1、将hive直接与mongodb关联，可以实时更新，方式也最简单。
  - 2、使用mongodump命令备份mongoDB数据表，之后上传到Hadoop的HDFS文件系统里，使用hive读入。
  - 3、使用mongoexport命令，导出数据库的单个表或者相关列。

- 前两种方法需要mongoDB的jar包，下载地址：

  http://mvnrepository.com/

  mongo-hadoop-core

  mongo-hadoop-hive

  mongo-java-driver

- 并且拷贝到$HIVE_HOME/lib和$HADOOP_HOME/share/hadoop/common里

- 具体方法可参考mongodb官方论坛：https://www.mongodb.com/blog/post/using-mongodb-hadoop-spark-part-2-hive-example

- 第三种，导出指定列：车牌、地点、与时间, 并导入HDFS中：

```
mongoexport -h 192.168.118.8 -d illegals -c illegal -f
place,plateNumber,createdAt -o ./Desktop/illegals/illegal.csv

hadoop fs -mkdir /illegals

hadoop fs -put ./Desktop/illegals/illegal.csv  /illegals/
```

- 命令行输入hive，在hive里create database illegals，use illegals

```
create table illegal(id string,plateNumber string,place string,createdAt string)
row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES("separatorChar"=",")
 STORED AS TEXTFILE;

load data inpath "/illegal/illegal.csv" into table illegal;
```

SerDe是Serialize/Deserilize的简称，
目的是用于序列化和反序列化。

WinStar 陈光跃

- 查询最多违法地点：

insert overwrite directory '/illegal/all_place.txt' select place,count(1) as count from illegal group by place order by count desc;

root@master:/home/cgy

```
pplication_1531791718901_0003/
Kill Command = /usr/hadoop/hadoop-3.0.3/bin/hadoop job  -kill job_1531791718901_
0003
Hadoop job information for Stage-1: number of mappers: 12; number of reducers: 1
2
2018-07-17 11:43:11,361 Stage-1 map = 0%,  reduce = 0%
2018-07-17 11:44:11,733 Stage-1 map = 0%,  reduce = 0%, Cumulative CPU 55.02 sec
2018-07-17 11:44:30,137 Stage-1 map = 14%,  reduce = 0%, Cumulative CPU 88.06 se
c
2018-07-17 11:44:31,158 Stage-1 map = 22%,  reduce = 0%, Cumulative CPU 90.94 se
c
2018-07-17 11:45:02,803 Stage-1 map = 25%,  reduce = 0%, Cumulative CPU 134.02 s
ec
2018-07-17 11:45:03,879 Stage-1 map = 31%,  reduce = 0%, Cumulative CPU 135.92 s
ec
2018-07-17 11:45:07,966 Stage-1 map = 33%,  reduce = 0%, Cumulative CPU 141.67 s
ec
2018-07-17 11:45:08,984 Stage-1 map = 39%,  reduce = 0%, Cumulative CPU 143.91 s
ec
2018-07-17 11:45:10,012 Stage-1 map = 44%,  reduce = 0%, Cumulative CPU 146.6 se
c
2018-07-17 11:45:14,106 Stage-1 map = 56%,  reduce = 0%, Cumulative CPU 151.11 s
ec
2018-07-17 11:45:15,146 Stage-1 map = 67%,  reduce = 1%, Cumulative CPU 154.64 s
```

root@master:/home/cgy

```
place":"108国道 45296
place":"工农路  47998
place":"204省道金鸡滩段（榆林交警三大队）        49456
place":"北二环朱宏路立交        49997
place":"碑林区其他道路  50238
place":"曲江大队其它路段        50444
place":"未央路凤城南路十字      51928
place":"包茂高速安靖段延安方向  55295
place":"东二环  57143
place":"未央路  58818
place":"西汉高速秦岭二号隧道出口09158100000     58900
place":"210国道 62510
place":"青银高速吴堡段刘家坪绥德方向    64279
place":"张家堡  66158
place":"长安中路        68248
place":"西汉高速大河坝09158100000       76620
place":"西汉高速安康段（查询电话09158100000;查询网址：www.akjjzd.gov.cn）0119公
里400米 91616
place":"西汉高速秦岭隧道（查询电话09158100000;查询网址：www.akjjzd.gov.cn）0000
公里000米       92677
place":"南二环  118390
place":"包南线(榆林段)  210929
Time taken: 270.094 seconds, Fetched: 353561 row(s)
hive>
```

WinStar 陈光跃

- 车辆违法高发地与数据清洗

insert overwrite directory '/illegal/car_place.csv' select plateNumber,place,count(1) as count from illegal group by plateNumber,place order by count desc;

```
2  plateNumber":"无SOH place":"东西七号路SOH 22223
3  plateNumber":"无SOH place":"东西八号路SOH 20751
4  plateNumber":"无SOH place":"108国道SOH 13837
5  plateNumber":"无SOH place":"东西四号路SOH 13325
6  plateNumber":"无SOH place":"秦户路SOH 13000
7  plateNumber":"无SOH place":"南北一号路SOH 8354
8  plateNumber":"无SOH place":"南北二号路SOH 7544
9  plateNumber":"无SOH place":"东西三号路SOH 7497
10 plateNumber":"无SOH place":"108国道1333公里SOH 7110
11 plateNumber":"无SOH place":"南北三号路SOH 6876
12 plateNumber":"无SOH place":"东西二号路SOH 5889
13 plateNumber":"无SOH place":"南北九号路SOH 5859
14 plateNumber":"无SOH place":"南北四号路SOH 5828
15 plateNumber":"无SOH place":"108国道0000公里SOH 5719
16 plateNumber":"无SOH place":"107省道0000公里SOH 5643
17 plateNumber":"无SOH place":"南北五号路SOH 5474
18 plateNumber":"无SOH place":"东西一号路SOH 5426
19 plateNumber":"无SOH place":"南北六号路SOH 5019
20 plateNumber":"无SOH place":"南北八号路SOH 4740
21 plateNumber":"无SOH place":"南北七号路SOH 4404
22 plateNumber":"无SOH place":"其他路口SOH 4281
23 plateNumber":"       SOH place":"渭清路SOH 3750
24 plateNumber":"无SOH place":"由老路SOH 3720
25 plateNumber":"无SOH place":"户祖路SOH 3448
26 plateNumber":"0000000SOH place":"108国道SOH 3281
27 plateNumber":"无SOH place":"西稍门SOH 2743
28 plateNumber":"无SOH place":"机场高速公路1公里SOH 2505
29 plateNumber":"陕ASOH place":"户祖路SOH 2355
30 plateNumber":"无SOH place":"人民路财政局十字SOH 1973
31 plateNumber":"无SOH place":"西三环未央段SOH 1966
32 plateNumber":"无SOH place":"电影院什字SOH 1904
33 plateNumber":"000000SOH place":"108国道SOH 1763
34 plateNumber":"无SOH place":"土门SOH 1740
35 plateNumber":"无SOH place":"广场西路SOH 1712
36 plateNumber":" 无SOH place":"东西四号路SOH 1558
37 plateNumber":"无SOH place":"204省道SOH 1550
38 plateNumber":"无SOH place":"107省道SOH 1546
```

WinStar 陈光跃

```python
import csv

myReader=csv.reader(open('/Users/pingguo/Desktop/car_place.csv',encoding='utf-8'))
#i=0
with open('/Users/pingguo/Desktop/car_place_clear.csv','w',newline='',encoding='utf-8') as csvfile:
    myWriter=csv.writer(csvfile)
    for row in myReader:
    #    i=i+1
     #   if i==100:
       #     break
        if row[0][14]=='无' or row[0][14]==' 'or row[0][14]=='0'or row[0][14]=='不'or row[0][14]=='*' or row[0][3]=='c'or row[0][16]=='\x01':
          #  i=i-1
            continue
        else:
            myWriter.writerow(row)
```


```
"plateNumber"":""陕        ]place"":""音乐学院以南@由南向北SOH 885"
"plateNumber"":""陕        ]place"":""音乐学院以南@由南向北SOH 663"
"plateNumber"":""陕        ]place"":""十天高速白河引线SOH 656"
"plateNumber"":""陕        ]place"":""长安路-陕师大向南@由北向南SOH 533"
"plateNumber"":""陕        ]place"":""长安路-陕师大向南@由北向南SOH 388"
"plateNumber"":""陕        ]place"":""县医院十字SOH 346"
"plateNumber"":""陕        ]place"":""音乐学院以南@由南向北SOH 320"
"plateNumber"":""陕        ]place"":""桑红路SOH 317"
"plateNumber"":""陕        ]place"":""长安路-陕师大向南@由北向南SOH 308"
"plateNumber"":""闽        ]place"":""航天大道东段SOH 299"
"plateNumber"":""陕        place"":""音乐学院以南@由南向北SOH 295"
"plateNumber"":""陕        place"":""红椿-东木SOH 287"
"plateNumber"":""陕        place"":""航天大道东段SOH 285"
"plateNumber"":""陕        place"":""塔哑公安卡口SOH 269"
"plateNumber"":""陕        place"":""迎宾大道SOH 266"
"plateNumber"":""陕        place"":""恒紫路SOH 242"
"plateNumber"":""陕        place"":""308省道28公里SOH 241"
"plateNumber"":""陕        place"":""南郑大道东段圣水路十字路口SOH 226"
"plateNumber"":""陕        place"":""街道－水码头0000公里SOH 218"
"plateNumber"":""陕        lace"":""建章路车辆厂十字@由南向北SOH 217"
"plateNumber"":""陕        place"":""街道－水码头0000公里SOH 214"
"plateNumber"":""陕        place"":""308省道28公里SOH 213"
"plateNumber"":""陕        place"":""老街1号路口SOH 213"
"plateNumber"":""陕        place"":""江北大道进站路什字SOH 209"
"plateNumber"":""陕        place"":""彭衙路SOH 208"
"plateNumber"":""陕        place"":""叩街路26公里960米SOH 197"
"plateNumber"":""陕        place"":""长安路-师大路十字SOH 195"
"plateNumber"":""陕        place"":""延长县严管一条街影剧院路段SOH 193"
"plateNumber"":""陕        place"":""农兴路全段SOH 192"
"plateNumber"":""陕        place"":""长安南路SOH 191"
"plateNumber"":""陕        place"":""东兴路南段SOH 187"
"plateNumber"":""陕        place"":""柞水迎宾大道SOH 184"
"plateNumber"":""陕        place"":""科技一路SOH 184"
"plateNumber"":""陕        place"":""体育馆南路SOH 182"
"plateNumber"":""陕        place"":""安（村）哑（柏）路蓝田段9公里540米SOH 1
"plateNumber"":""陕        place"":""省203－小河口村0000公里SOH 179"
"plateNumber"":""陕        place"":""316国道宗营段SOH 178"
```

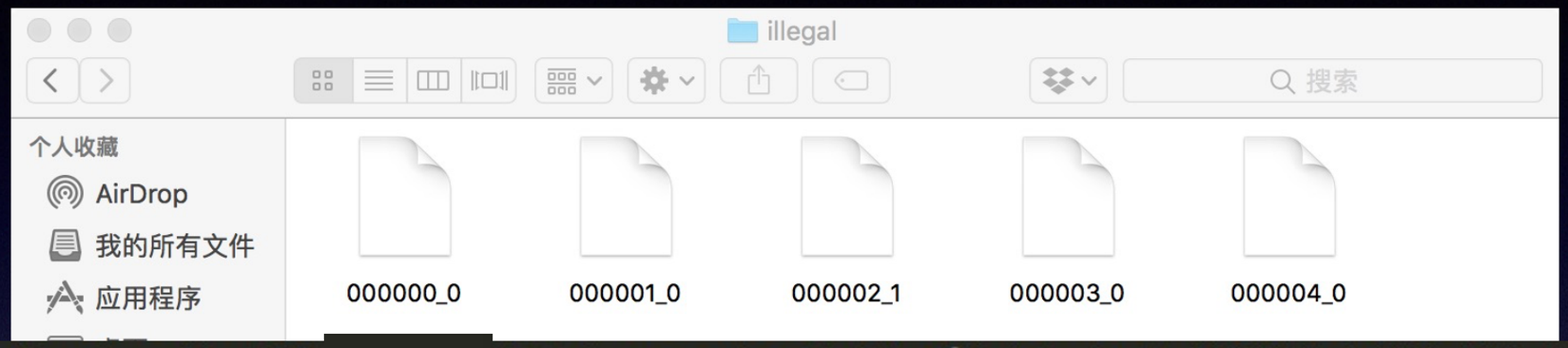

其实这里更好的方法是用后面将要介绍的高级hive用法，因为写这个脚本的时候还没了解，另外中文正则化也可以考虑一下

carPlace_clear

WinStar 陈光跃

insert overwrite directory '/illegal/' select plateNumber,createdAt from illegal ;

车辆时间




```
9   plateNumber":"陕 SOH createdAt":{"$date":"2017-03-01T02:05:00.000Z"}}
10  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T02:00:00.000Z"}}
11  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T02:20:00.000Z"}}
12  plateNumber":"冀 SOH createdAt":{"$date":"2017-03-02T02:22:00.000Z"}}
13  plateNumber":"陕 SOH createdAt":{"$date":"2016-07-22T04:42:49.000Z"}}
14  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T04:55:00.000Z"}}
15  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T06:47:00.000Z"}}
16  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T06:53:00.000Z"}}
17  plateNumber":"陕 SOH createdAt":{"$date":"2017-02-18T10:30:00.000Z"}}
18  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-01T09:00:00.000Z"}}
19  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T05:12:00.000Z"}}
20  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T02:15:00.000Z"}}
21  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T04:35:00.000Z"}}
22  plateNumber":"不 createdAt":{"$date":"2017-03-02T02:20:00.000Z"}}
23  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T07:20:00.000Z"}}
24  plateNumber":"陕 SOH createdAt":{"$date":"2017-03-02T07:52:00.000Z"}}
```

使用mapReduce模型进行分析

```
public class TimeCount {
    public TimeCount() {
    }
    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> [<in>...] <out>");
            System.exit(2);
        }
        Job job = Job.getInstance(conf, "CarTime count");
        job.setJarByClass(TimeCount.class);
        job.setMapperClass(TimeCount.CarTimeMapper.class);
        job.setCombinerClass(TimeCount.TimeSumReducer.class);
        job.setReducerClass(TimeCount.TimeSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        for(int i = 0; i < otherArgs.length - 1; ++i) {
            FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
        }
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }

    public static class TimeSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();
        public TimeSumReducer() {
        }
        public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text, IntWritable>.Context context) throws
IOException, InterruptedException {
            int sum = 0;

            IntWritable val;
            for(Iterator carTime = values.iterator(); carTime.hasNext(); sum += val.get()) {
                val = (IntWritable)carTime.next();
            }

            this.result.set(sum);
            context.write(key, this.result);
        }
    }
                this.carTime.set(this.plate +this.time+",");
                context.write(this.carTime,one);
            }

        }
    }
}
```

WinStar 陈光跃

```java
    public static class CarTimeMapper extends Mapper<Object, Text, Text, IntWritable> {
        private static final IntWritable one = new IntWritable(1);
        private String plate ;
        private String time ;
        private Text carTime = new Text();
        public CarTimeMapper() {
        }

        public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context context) throws IOException, InterruptedException
{


            String line = value.toString();
            if(line.length()<=15){
                return;
            }
            if (line.charAt(14) == '无' || line.charAt(14) == ' ' || line.charAt(14) >= '0'&&line.charAt(14) <= '9' || line.charAt(14) == '不'
                    || line.charAt(14) == '*'|| line.charAt(14) == '?'|| line.charAt(14) == '='|| line.charAt(14) == '-'|| line.charAt(14) ==
'.'|| line.charAt(14) == '/') {
                System.out.print("wrong plate number\n");
            } else {
                try {
                    StringTokenizer itr = new StringTokenizer(line, "\":{$");
                    itr.nextToken();
                    this.plate = itr.nextToken().replaceFirst("createdAt","");
                    itr.nextToken();
                    int hours = Integer.parseInt(itr.nextToken().substring(11, 13));
                    if (hours > 0 && hours < 7) {
                        this.time = ",夜间00-07";
                    } else if (hours < 11) {
                        this.time = ",早高峰07-11";
                    } else if (hours < 14) {
                        this.time = ",午间11-14";
                    } else if (hours < 17) {
                        this.time = ",下午14-17";
                    } else if (hours < 20) {
                        this.time = ",晚高峰17-20";
                    } else {
                        this.time =  ",夜间20-24";
                    }

                } catch (Exception e) {
                    System.out.print("time change exception or data wrong exception"+e+'\n');
                     return;
                }
            this.carTime.set(this.plate +this.time+",");
            context.write(this.carTime,one);

        }

    }
}
```

WinStar 陈光跃

```
5157535   陕A          ,夜间00-07,  1
5157536   陕A          ,早高峰07-11, 1
5157537   陕A          夜间00-07,   1
5157538   陕A          ,夜间00-07,  1
5157539   陕A          ,夜间00-07,  1
5157540   陕A          ,夜间00-07,  1
5157541   陕A          ,夜间00-07,  1
5157542   陕A          ,夜间00-07,  1
5157543   陕A          ,夜间00-07,  2
5157544   陕A          ,早高峰07-11, 1
5157545   陕A          ,夜间00-07,  3
5157546   陕A          ,早高峰07-11, 3
5157547   陕A          ,夜间00-07,  1
5157548   陕A          ,早高峰07-11, 1
5157549   陕A          ,夜间00-07,  1
5157550   陕A          ,早高峰07-11, 1
5157551   陕A          ,早高峰07-11, 3
5157552   陕A          ,夜间00-07,  3
5157553   陕A          ,早高峰07-11, 2
5157554   陕A          ,早高峰07-11, 3
5157555   陕A          ,夜间00-07,  1
5157556   陕A          ,早高峰07-11, 1
5157557   陕A          ,夜间00-07,  1
5157558   陕A          ,早高峰07-11, 1
5157559   陕A          ,午间11-14,  1
5157560   陕A          ,午间11-14,  1
5157561   陕A          ,夜间00-07,  11
5157562   陕A          ,早高峰07-11, 12
5157563   陕A          ,夜间00-07,  1
5157564   陕A          ,早高峰07-11, 2
5157565   陕A          ,夜间00-07,  1
5157566   陕A          ,午间11-14,  1
5157567   陕A          ,夜间00-07,  1
5157568   陕A          ,早高峰07-11, 3
5157569   陕A          ,下午14-17,  2
5157570   陕A          ,午间11-14,  2
5157571   陕A          ,夜间00-07,  11
5157572   陕A          ,夜间20-24,  1
5157573   陕A          ,早高峰07-11, 15
```

WinStar 陈光跃

总结、Hive与大数据系统其他部分的比较

MapReduce的程序写起来很繁琐，为了简化这个过程，于是就有了Pig和Hive。Pig是接近脚本方式去描述MapReduce，Hive则用的是SQL。它们把脚本和SQL语言翻译成MapReduce程序，丢给计算引擎去计算，而你就从繁琐的MapReduce程序中解脱出来，用更简单更直观的语言去写程序了。

Hive有巨大的优势。一个是它太容易写了。词频统计，用SQL描述就只有一两行，MapReduce写起来大约要几十上百行。Hive逐渐成长成了大数据仓库的核心组件。甚至很多公司的流水线作业集完全是用SQL描述，因为易写易改，一看就懂，容易维护。

缺点：Hive在MapReduce上跑，很慢。于是Impala，Presto，Drill诞生了。三个系统的核心理念是，MapReduce引擎太慢，因为它太通用，太强壮，太保守，我们SQL需要更轻量，更激进地获取资源，更专门地对SQL做优化，而且不需要那么多容错性保证（因为系统出错了大不了重新启动任务，如果整个处理时间更短的话，比如几分钟之内）。这些系统让用户更快速地处理SQL任务，牺牲了通用性稳定性等特性。如果说MapReduce是大砍刀，砍啥都不怕，那上面三个就是剔骨刀，灵巧锋利，但是不能搞太大太硬的东西。

如果追求实时性，实时计算storm与准实时计算spark更加受欢迎。

Hive对表的优化管理
分区与分桶

```
CREATE TABLE logs (ts BIGINT, line STRING)
PARTITIONED BY (dt STRING, country STRING);
```

```
LOAD DATA LOCAL INPATH 'input/hive/partitions/file1'
INTO TABLE logs
PARTITION (dt='2001-01-01', country='GB');
```

分区：一种根据分区列的值，将表划分为粗粒度部分的方法，减少每一次扫描总数据量，提升性能。

WinStar 陈光跃

```
/user/hive/warehouse/logs
├── dt=2001-01-01/
│   ├── country=GB/
│   │   ├── file1
│   │   └── file2
│   └── country=US/
│       └── file3
└── dt=2001-01-02/
    ├── country=GB/
    │   └── file4
    └── country=US/
        ├── file5
        └── file6
```

分区：一种根据分区列的值将表划分为粗粒度部分的方法，减少每一次扫描总数据量，提升性能。

```
hive> SHOW PARTITIONS logs;
dt=2001-01-01/country=GB
dt=2001-01-01/country=US
dt=2001-01-02/country=GB
dt=2001-01-02/country=US
```

show partitions可以查看分区

WinStar 陈光跃

分桶

```
CREATE TABLE bucketed_users (id INT, name STRING)
CLUSTERED BY (id) INTO 4 BUCKETS;
```

分桶是通常是通过对指定列进行哈希计算来实现的，通过哈希值将一个列名下的数据切分为一组桶，并使每个桶对应于该列名下的一个存储文件。

WinStar 陈光跃

```
hive> SELECT * FROM users;
0    Nat
2    Joe
3    Kay
4    Ann
```

To populate the bucketed table, we need to set the `hive.enforce.bucketing` property to `true` so that Hive knows to create the number of buckets declared in the table definition. Then it is just a matter of using the `INSERT` command:

```
INSERT OVERWRITE TABLE bucketed_users
SELECT * FROM users;
```

```
hive> dfs -ls /user/hive/warehouse/bucketed_users;
```

```
000000_0
000001_0
000002_0
000003_0
```

WinStar 陈光跃

目标

可以有选择的去读表里的数据行

```
hive> SELECT * FROM bucketed_users
    > TABLESAMPLE(BUCKET 1 OUT OF 4 ON id);
4     Ann
0     Nat


hive> SELECT * FROM bucketed_users
    > TABLESAMPLE(BUCKET 1 OUT OF 2 ON id);
4     Ann
0     Nat
2     Joe
```

分区与分桶比较

在分区数量过于庞大以至于可能导致文件系统崩溃时，我们就需要使用分桶来解决问题了。分区中的数据可以被进一步拆分成桶，不同于分区对列直接进行拆分，桶往往使用列的哈希值对数据打散，并分发到各个不同的桶中从而完成数据的分桶过程。每个桶中的数据条数不一定相等。在数据量足够大的情况下，分桶比分区，更高的查询效率。

WinStar 陈光跃

储存格式：storage format

管理Hive中的表存储有两个维度：行格式和文件格式。 行格式指示如何存储行和特定行中的字段。 在Hive的说法中，行格式由SerDe定义，SerDe是Serializer-Deserializer的portmanteau字。
当作为反序列化器（查询表时）时，SerDe会将文件中字节的一行数据反序列化为Hive内部使用的对象，以对该行数据进行操作。
文件格式规定了行中字段的容器格式。 最简单的格式是纯文本文件，但也有面向行和列导向的二进制格式。

Hive的高级使用1
—MapReduce脚本

```
#!/usr/bin/env python

import re
import sys

for line in sys.stdin:
  (year, temp, q) = line.strip().split()
  if (temp != "9999" and re.match("[01459]", q)):
    print "%s\t%s" % (year, temp)
```

We can use the script as follows:

```
hive> ADD FILE /Users/tom/book-workspace/hadoop-book/ch17-hive/
src/main/python/is_good_quality.py;
hive> FROM records2
    > SELECT TRANSFORM(year, temperature, quality)
    > USING 'is_good_quality.py'
    > AS year, temperature;
```

WinStar 陈光跃

Hive的高级使用2
—自定函数UDF

```java
package com.hadoopbook.hive;

import org.apache.commons.lang.StringUtils;
import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.Text;

public class Strip extends UDF {
  private Text result = new Text();

  public Text evaluate(Text str) {
    if (str == null) {
      return null;
    }
    result.set(StringUtils.strip(str.toString()));
    return result;
  }  public Text evaluate(Text str, String stripChars) {
    if (str == null) {
      return null;
    }
    result.set(StringUtils.strip(str.toString(), stripChars));

  return result;
  }
}
```

WinStar 陈光跃

UDF必须是org.apache.hadoop.hive.ql.exec.UDF的子类.

A UDF must implement at least one evaluate() method.

```
CREATE FUNCTION strip AS 'com.hadoopbook.hive.Strip'
USING JAR '/path/to/hive-examples.jar';
```

```
hive> SELECT strip('  bee  ') FROM dummy;
bee
hive> SELECT strip('banana', 'ab') FROM dummy;
nan
```

WinStar 陈光跃

Hive的高级使用2
—自定聚合函数UDAF

```
package com.hadoopbook.hive;

import org.apache.hadoop.hive.ql.exec.UDAF;
import org.apache.hadoop.hive.ql.exec.UDAFEvaluator;
import org.apache.hadoop.io.IntWritable;

public class Maximum extends UDAF {

  public static class MaximumIntUDAFEvaluator implements UDAFEvaluator {

    private IntWritable result;

    public void init() {
      result = null;
    }
```

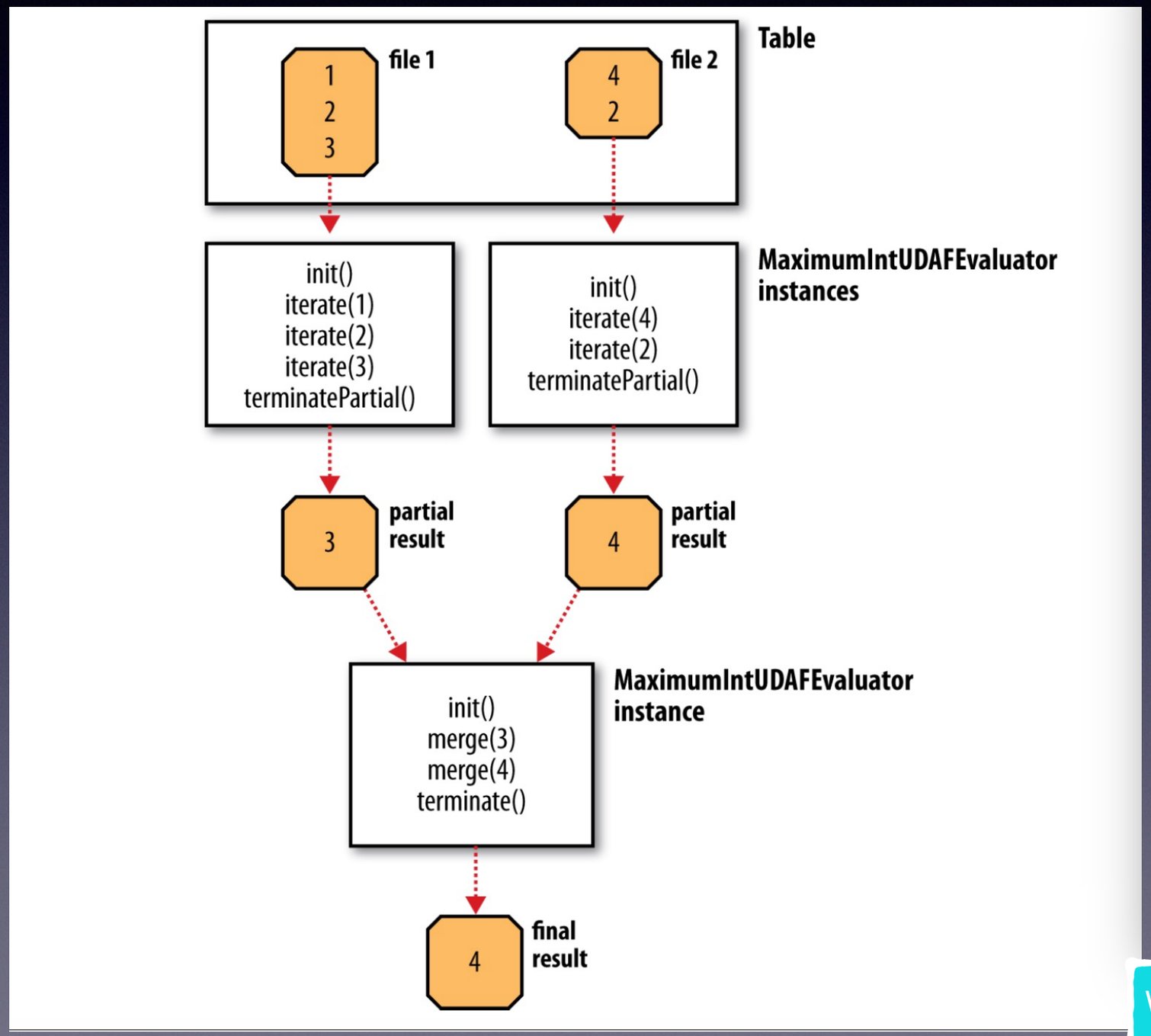
WinStar 陈光跃

```java
    public boolean iterate(IntWritable value) {
      if (value == null) {
        return true;
      }
      if (result == null) {
        result = new IntWritable(value.get());
      } else {
        result.set(Math.max(result.get(), value.get()));
      }
      return true;
    }

    public IntWritable terminatePartial() {
      return result;
    }

    public boolean merge(IntWritable other) {
      return iterate(other);
    }

    public IntWritable terminate() {
      return result;
    }
  }
}
```

WinStar 陈光跃

UDAF的使用

```
hive> CREATE TEMPORARY FUNCTION maximum AS 'com.hadoopbook.hive.Maximum';
hive> SELECT maximum(temperature) FROM records;
```



WinStar 陈光跃