

Hive (20:30准时开课)

--讲师: Yasaka

3月12日周末班 2月26日全日制班 3月19日 在线班 欢迎您的到来!

需要代码、PPT、视频等资料请加以下几位老师QQ:

贾老师: 1786418286

何老师: 1926106490

詹老师: 2805048645

讨论技术可以加入以下QQ群: 172599077 , 156927834

热烈庆祝1221全日制班以及1226周末班爆满开班!!!

16.1.1日之后学费上调, 考虑春节后培训的学生可以提前报名, 预订座位, 不管以后何时过来学习, 费用都是以报名进的费用为主!



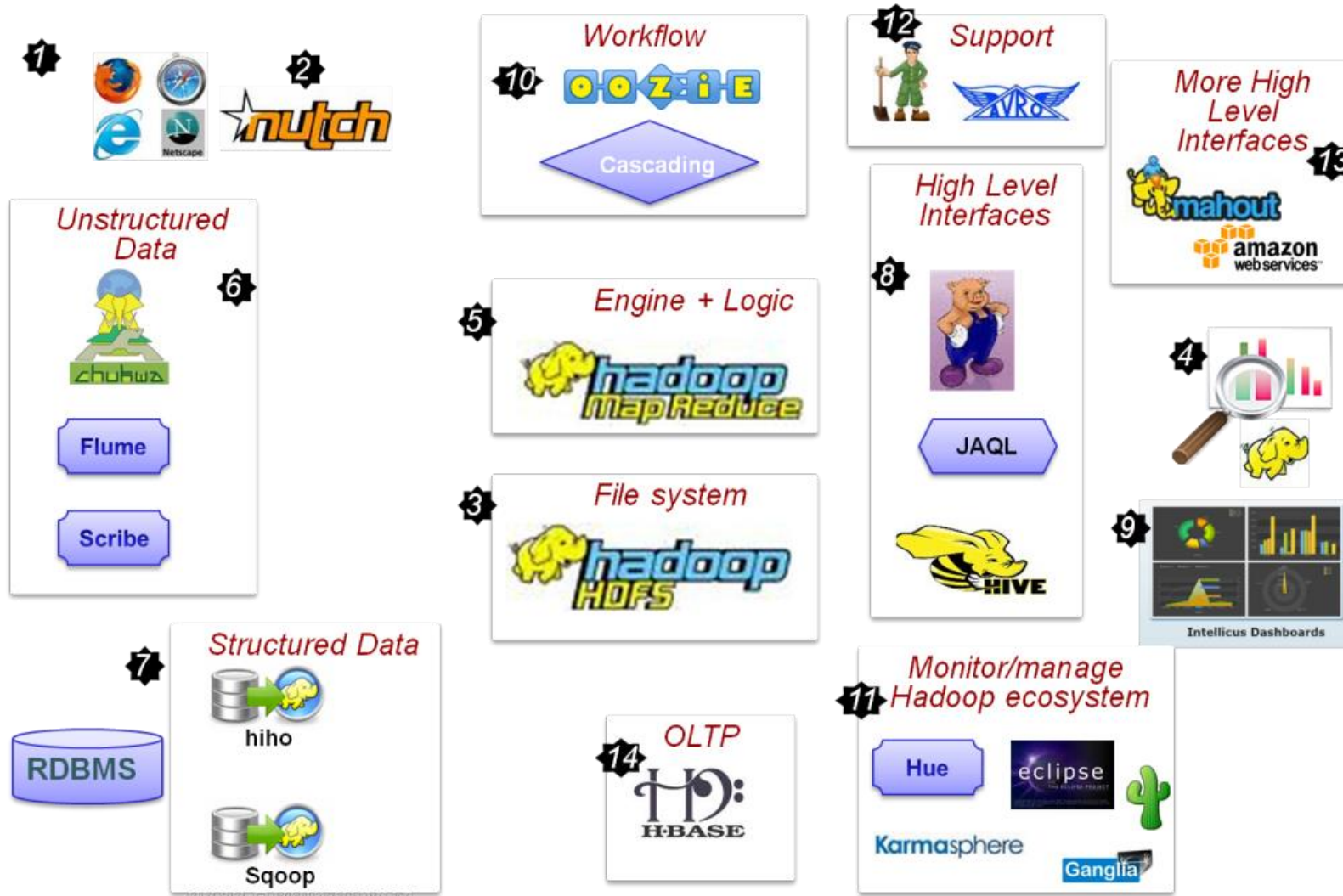
- Hive可以做什么
- 学习Hive有什么用
- Hive到底是什么及组件有什么
- 把Hive安装起来
- Hive的元信息存储
- Hive建表
- Hive加载数据
- Hive执行命令方式



- Hive创建数据库及切换
- Hive数据类型
- Hive脚本
- Hive分区表
- Hive数据操作
- Hive函数操作
- Hive自定义函数
- Hive结合Zookeeper支持锁功能



Hadoop Ecosystem Map



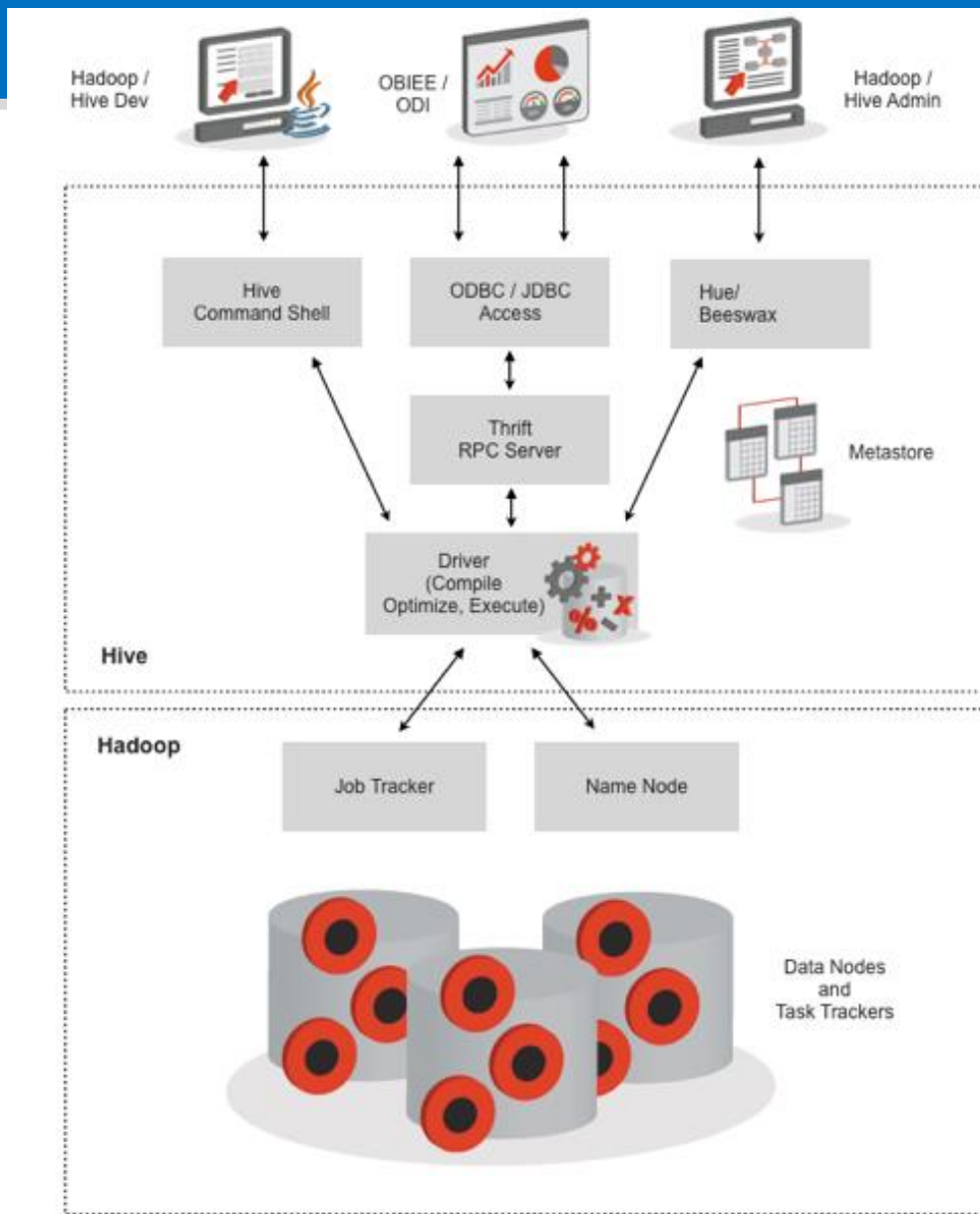
- Hive是Hadoop生态系统中必不可少的一个工具！
- 它提供了一个SQL（结构化查询语言）方言！
- 可以查询存储在Hadoop分布式文件系统（HDFS）中的数据或其他和Hadoop集成的文件系统，如MapR-FS、Amazon的S3和像HBase（Hadoop数据库）和Cassandra这样的数据库中的数据！

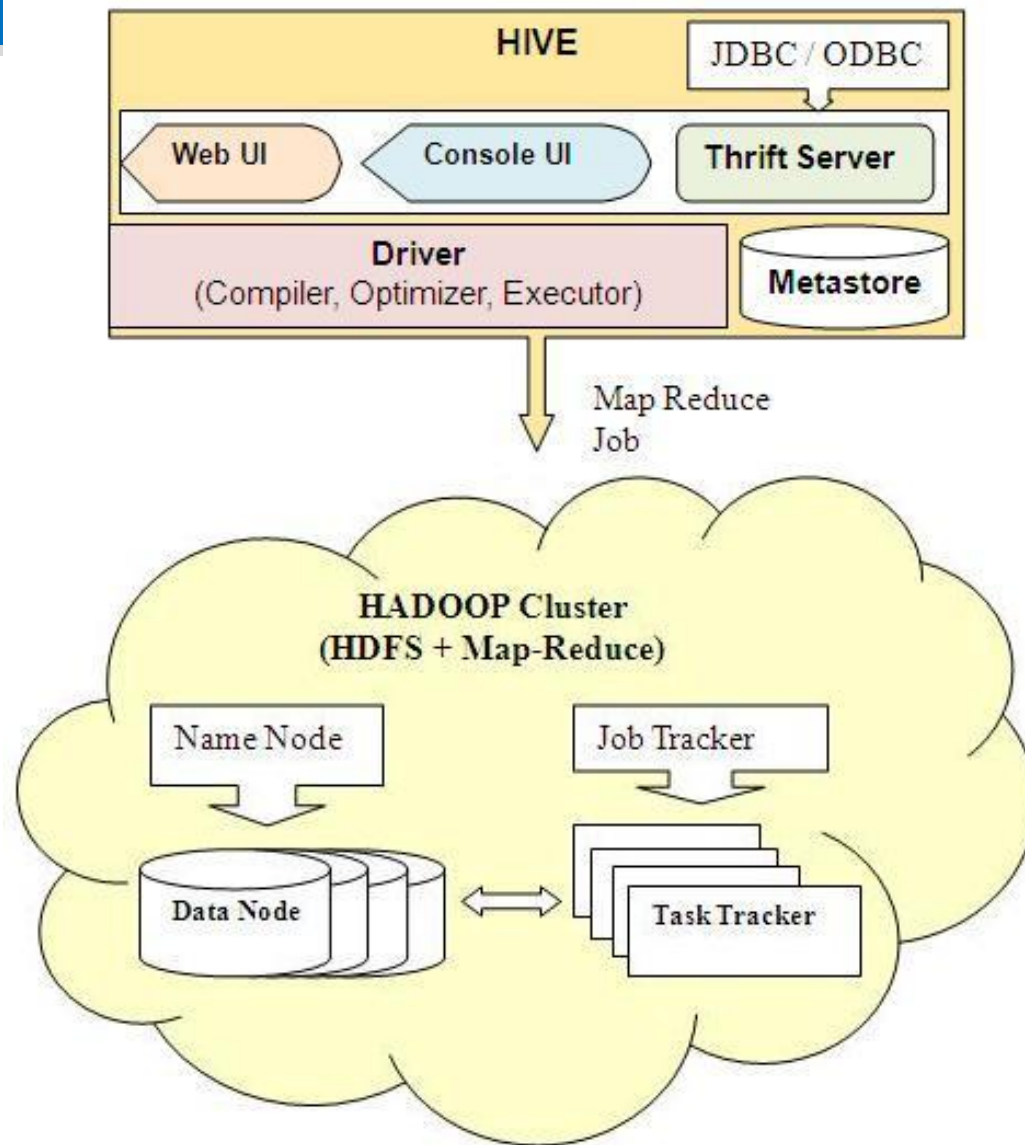


- 大多数数据仓库应用程序都是使用关系数据库进行实现的，并使用SQL作为查询语言，Hive降低了将这些应用程序转移到Hadoop系统上的难度！
- 凡是会使用SQL语言的开发人员都可以很轻松的学习并使用Hive！
- 如果没有Hive我们将面临一个艰巨的挑战就是如何将他们的SQL应用程序移植到Hadoop上！
- 很多时候公司里面的SQL专家、数据库设计人员、业务分析师在大量使用Hive！
- 帮助Hadoop开发人员对Hive进行调优和定制！



Hive是什么





- 最新版本HIVE-1.2.1
- vi ./bin/hive-config.sh
- 添加环境变量
- export JAVA_HOME=/usr/soft/jdk1.7.0_71
- export HADOOP_HOME=/usr/hadoopsoft/hadoop-2.5.2
- export HIVE_HOME=/usr/hadoopsoft/apache-hive-1.2.1-bin



- 编辑hive-site.xml
- cp conf/hive-default.xml.template conf/hive-site.xml

```
<property>
  <name>hive.metastore.warehouse.dir</name>
  <value>/user/hive/warehouse</value>
</property>
<property>
  <name>hive.exec.scratchdir</name>
  <value>/tmp/hive</value>
</property>
<property>
  <name>hive.exec.local.scratchdir</name>
  <value>/tmp/hive</value>
  <description>Local scratch space for Hive jobs</description>
</property>
<property>
  <name>hive.downloaded.resources.dir</name>
  <value>/tmp/hive/resources</value>
  <description>Temporary local directory for added resources in the remote file system.</description>
</property>
```



- 配置hive的log4j: `cp conf/hive-log4j.properties.template conf/hive-log4j.properties`
- `#log4j.appender.EventCounter=org.apache.hadoop.hive.shims.HiveEventCounter`
- `log4j.appender.EventCounter=org.apache.hadoop.log.metrics.EventCounter`
- 否则会有警告：
- `WARN conf.HiveConf: HiveConf of name hive.metastore.local does not exist`
- `WARNING: org.apache.hadoop.metrics.jvm.EventCounter is deprecated. Please use org.apache.hadoop.log.metrics.EventCounter in all the log4j.properties files.`



- 启动时候遇到错误：

```
[ERROR] Terminal initialization failed; falling back to unsupported
java.lang.IncompatibleClassChangeError: Found class jline.Terminal, but interface was expected
    at jline.TerminalFactory.create(TerminalFactory.java:101)
    at jline.TerminalFactory.get(TerminalFactory.java:158)
    at jline.console.ConsoleReader.<init>(ConsoleReader.java:229)
    at jline.console.ConsoleReader.<init>(ConsoleReader.java:221)
    at jline.console.ConsoleReader.<init>(ConsoleReader.java:209)
    at org.apache.hadoop.hive.cli.CliDriver.setupConsoleReader(CliDriver.java:787)
    at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:721)
    at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:681)
    at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:621)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:606)
    at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
```



- 发现是jline有问题，最后找到，原来hadoop里面的是jline0.9.94 hive里面的是jline2.12
- 把/usr/hadoopsoft/apache-hive-1.2.1-bin/lib/ 里面的jline2.12放到hadoop中 /usr/hadoopsoft/hadoop-2.5.2/share/hadoop/yarn/lib/替换掉原来的jline0.9.94就可以成功了！



- 注：如果hadoop-env.sh中没有增加HADOOP_CLASSPATH的话会报下面的错误：
- java.lang.NoClassDefFoundError:
org/apache/hadoop/hive/ql/CommandNeedRetryException
- 修改\$HADOOP_HOME/conf/hadoop-env.sh
- #增加HADOOP_CLASSPATH
- export
HADOOP_CLASSPATH=.:\$CLASSPATH:\$HADOOP_CLASSPATH:\$HADOOP_H
OME/bin
- #记得修改完成以后，要将修改后的文件同步拷贝到其他的节点。



- 为了一次性成功，在hive主目录下找到conf文件夹下的hive_env.sh，将其中得HADOOP_HOME和HIVE_CONF_DIR放开并怕配置
- # Set HADOOP_HOME to point to a specific hadoop install directory
- export HADOOP_HOME=/usr/hadoopsoft/hadoop-2.5.2
- # Hive Configuration Directory can be controlled by:
- export HIVE_CONF_DIR=/usr/hadoopsoft/apache-hive-1.2.0-bin/conf



- 安装后启动./bin/hive
- 默认使用的是derby

```
<property>  
  <name>javax.jdo.option.ConnectionURL</name>  
  <value>jdbc:derby;;databaseName=metastore_db;create=true</value>  
  <description>JDBC connect string for a JDBC metastore</description>  
</property>
```

- 公司里更多是使用mysql存储元信息
- 找到一台机器安装MySQL服务器，spark003安装
- yum install mysql-server
- 启动mysql，在Windows安装一个mysql也是可以的，反正Hive连接mysql都是通过jdbc去连接的，所以Oracle也可以
- service mysqld start



- 安装用过MySQL的人就知道，实际上这样安装MySQL它只允许本地登录的！它不允许外面的去连接
- mysql
- use mysql
- select * from user;
- 记录了mysql用户的表，localhost第一个字段表示登陆的主机，然后是用户名密码



- 保证JDBC外面的用户也能连接的上,授权添加用户
- `grant all on *.* to root@'%' identified by '123456';`
- 手动创建数据库用于给hive存放元信息
- `create database hive_meta_data;`
- 放入mysql-connector-java-3.1.14-bin.jar到/usr/hadoopsoft/apache-hive-1.2.1-bin/lib/
- 配置连接关系型数据库4要素：

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://spark003:3306/hive_meta_data</value>
  <description>JDBC connect string for a JDBC metastore</description>
</property>
```



- 配置连接关系型数据库4要素：

```
<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.jdbc.Driver</value>
  <description>Driver class name for a JDBC metastore</description>
</property>
```

```
<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>root</value>
  <description>Username to use against metastore database</description>
</property>
```

```
<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>123456</value>
  <description>password to use against metastore database</description>
</property>
```



- 建表,Hive有自己的数据类型,表要和数据格式对应起来的
- create table alibaba_log (itemid string, userid string, action string, vtime string) row format delimited fields terminated by ',' stored as textfile ;
- 去/user/hive/warehouse下可以看到目录alibaba_log , 所以每个表对应HDFS上有个目录



- load data local inpath '/usr/hadoopsoft/apache-hive-1.2.1-bin/hive-alibaba-log.csv' into table alibaba_log;
- 可以到HDFS上看见目录下就有这个CSV文件，没有更改完全就直接放进去了
- select * from alibaba_log limit 3;
- 上面没有启动mapreduce，下一个查询你就会看见熟悉的mapreduce任务,也就是可以去spark001:8088页面里面看到Job
- select count(*) from alibaba_log;



- ./bin/hive -H
 - -e后面直接可以接一个Sql语句，一次性查询
 - -e <quoted-query-string> SQL from command line
 - -f后面接脚步文件
 - -f <filename> SQL from files
 - -i 接初始化文件
-
- 交互式不适合做批处理，所以我们都是在文件里写sql语句，通过一个hive -f就可以执行了



- 创建表是重点，还有重点是创建函数
- 数据库也可以创建，`CREATE DATABASE human_resources;`
- `USE human_resources` 切换当前数据库
- `hive -e 'show databases'`
- 默认是default库，`set hive.cli.print.current.db=true;`
- 描述表
- `desc alibaba_log;`



- 看官网学习

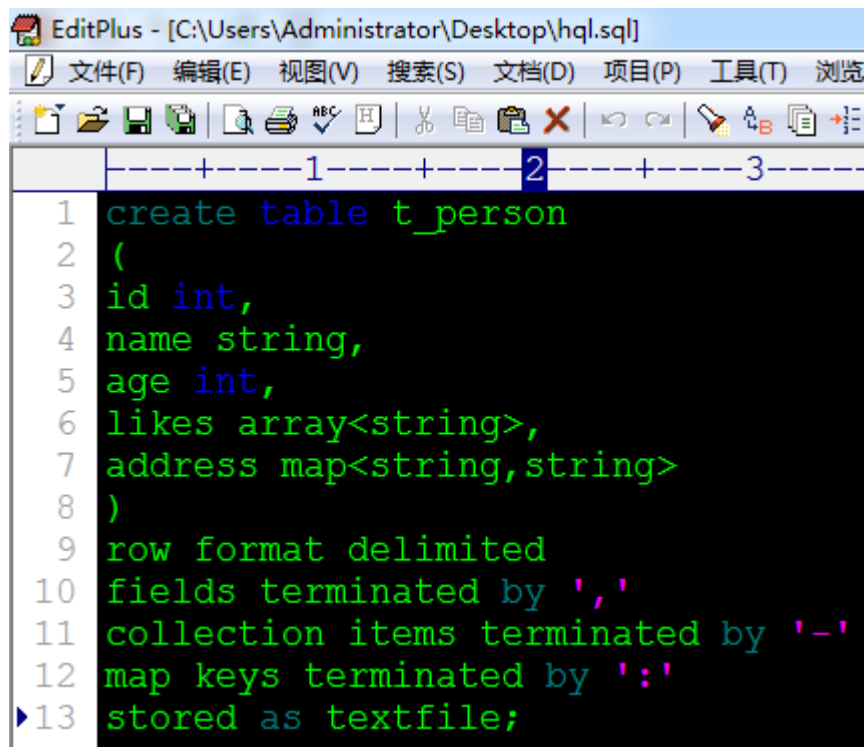
<https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL>

```
data_type
: primitive_type
| array_type
| map_type
| struct_type
| union_type  -- (Note: Available in Hive 0.7.0 and later)

primitive_type
: TINYINT
| SMALLINT
| INT
| BIGINT
| BOOLEAN
| FLOAT
| DOUBLE
| STRING
| BINARY      -- (Note: Available in Hive 0.8.0 and later)
| TIMESTAMP  -- (Note: Available in Hive 0.8.0 and later)
| DECIMAL    -- (Note: Available in Hive 0.11.0 and later)
| DECIMAL(precision, scale) -- (Note: Available in Hive 0.13.0 and later)
| DATE       -- (Note: Available in Hive 0.12.0 and later)
| VARCHAR   -- (Note: Available in Hive 0.12.0 and later)
| CHAR      -- (Note: Available in Hive 0.13.0 and later)
```



- `./bin/hive -f hql.sql`



```
1 create table t_person
2 (
3 id int,
4 name string,
5 age int,
6 likes array<string>,
7 address map<string,string>
8 )
9 row format delimited
10 fields terminated by ','
11 collection items terminated by '-'
12 map keys terminated by ':'
13 stored as textfile;
```



- 创建一个数据
- vi person.txt
- 1,zs,31,book-sports,city:changsha-street:xisanqi-zipcode:100010
- 1,ls,30,book,city:beijing-street:xisanqi-zipcode:100010
- 导入数据
- load data local inpath './person.txt' into table t_person;
- select * from t_person;
- 查询字段中的键
- select address['city'] from t_person where name='zs';
- 光查不做统计分析的话它不会用mapreduce



- 分区表把数据分成一块一块的，提高查询统计速度
- 按天来统计数据，vi post.txt
- load data local inpath './post.txt' into table t_log;
- 分区表导入需要指定分区
- load data local inpath './post.txt' into table t_log partition (day='2015-09-03');
- load data local inpath './post2.txt' into table t_log partition (day='2015-10-03');
- 我们可以看到HDFS上面有两个目录
- select count(*) from t_log where day='2015-09-03';
- 这样统计的时候就减少了输入的数据量，可以提高执行的速度
- 统计月？？？

```
110,1,100,2015-09-03
111,1,100,2015-09-03
112,1,100,2015-09-03
113,1,100,2015-09-03
```



- Hive Data Manipulation Language
- DML是对数据进行增删改，和关系型数据库是有区别的，是不能改的
- 为什么呢？数据在HDFS上的，在HDFS上官方是不提供修改，只提供追加的，追加还是用insert
- delete是删除数据
- <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DML#LanguageManualDML-Loadingfilesintotables>



- LOAD
- 上传文件到HDFS
- `hdfs dfs -put ./SogouQ.mini /user/root/`
- 导入到HIVE里面
- `load data inpath '/user/root/SogouQ.mini' into table t_log partition (day='2015-10-01');`
- 导入后会发现之前的文件所在的地方不存在了，存放数据的目录变了，block的位置还是没有变化的，所以对于HDFS本身来说没有动，只是改了个namenode的元数据，在fsimage或者内存中
- `/user/root/hive/warehouse/t_log/day=2015-10-01`



- INSERT
- insert into table t_log partition (day='2015-10-03') values ('999',2,'101','2012-09-09');
- 当成MR去执行，因为本来我们插入的数据是不存在的，不在文件中，实际上是通过MR将数据上传到HDFS里，产生一个新的文件
- 在目录下/user/root/hive/warehouse/t_log/day=2015-10-03
- 所以不建议用，会了也不用，因为使用MR插入数据非常慢的，所以使用LOAD插入数据
- DELETE
- 删除实际上都不用讲，你不在HIVE中删表，直接在HDFS删文件会吗？因为你HIVE中删数据就是HDFS上删文件



- SELECT
- 查询，HIVE侧重点就在这个地方了，查询和SQL完全集合一模一样
- *
- limit
- where (and or)
- distinct
- <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Select#LanguageManualSelect-PartitionBasedQueries>
- partition的东西当成一个字段就可以了
- having包含什么什么的
- REGEX还可以支持正则表达式



- <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF>
- HIVE里面的操作符和用户定义的函数UDF
- 内置操作符
- like也是内置操作符，其实是模糊查询的，和SQL里面一样，效率很低的，解决like的办法是索引，使用lucence和solr，解决like效率低的问题
- 数值运行操作符
- 逻辑操作符
- in是不是也是，看看在不在集合中
- HIVE中还有复杂类型数据类型，map,struct,array，这个是SQL中所没有的



- 内建函数
- count()
- round()
- 以后大家要知道去哪里去找！
- 类型转换的函数
- 日期转换的函数
- bigint意味着距离1970年1月1日
- current_date
- current_timestamp
- date_format(date/timestamp/string ts, string fmt)
- 逻辑函数
- 字符串函数
- 加密函数
- 上述函数和SQL里面是一样的找到函数然后给函数传参就可以了

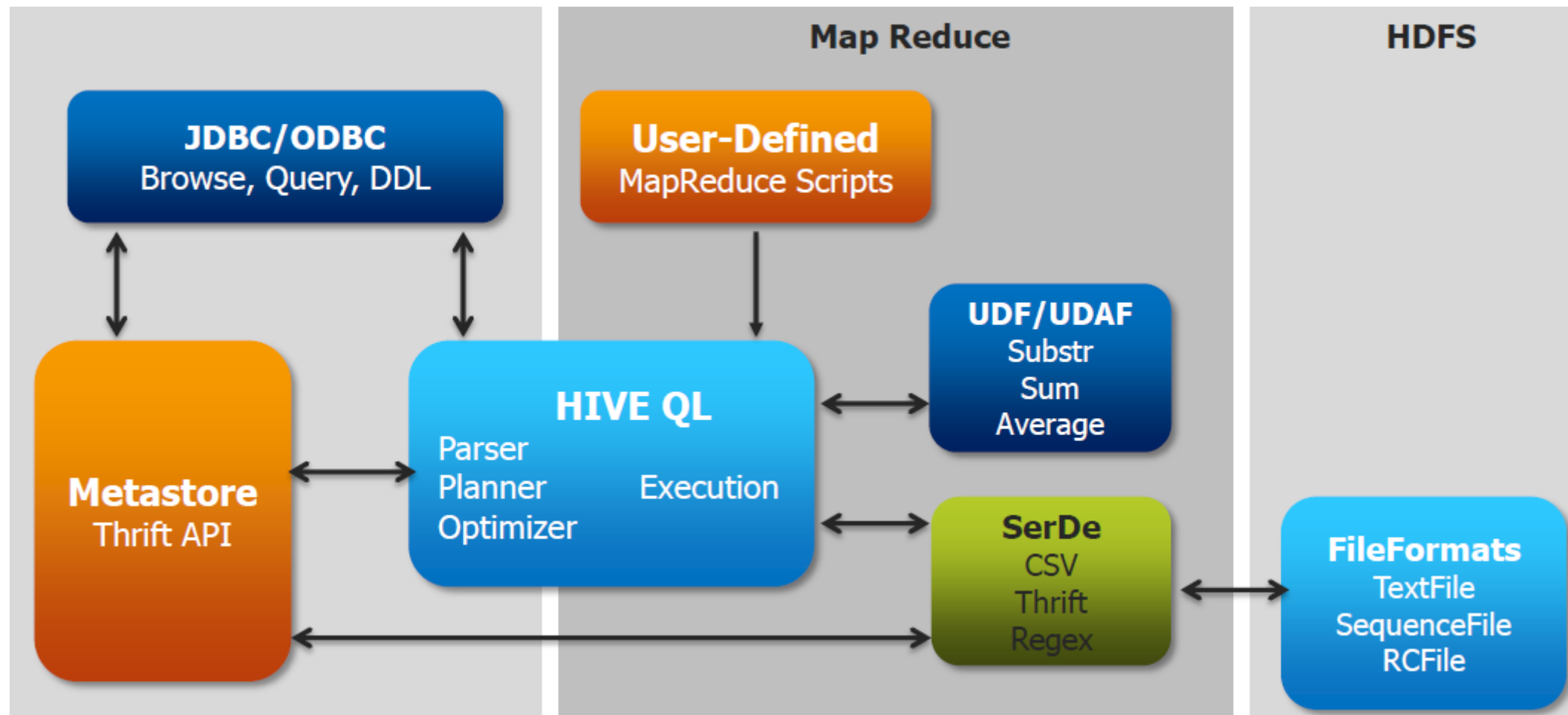


- 我们讲用户自定义的函数
- <https://cwiki.apache.org/confluence/display/Hive/HivePlugins>
- 刚有没有看见一个字符串变为日期的？
- 那我们要自己写JAVA代码来定义！
- 加入hadoop的jar包，加入hive的jar包
- 写好了UDF后选择要导出的类打包上传到服务器
- 到hive里面添加JAR包到class_path下
- add jar ./std.jar;
- create function std as 'com.bjsxt.hive.StringToDate';
- select std('20150909 122215', 'yyyyMMdd HH:mm:ss') from t_log;




```
StringToDate.java ✖
1 package com.bjsxt.hive;
2
3 import java.text.SimpleDateFormat;
4 import java.util.Date;
5
6 import org.apache.hadoop.hive.ql.exec.UDF;
7 import org.apache.hadoop.hive.serde2.io.TimestampWritable;
8 import org.apache.hadoop.io.Text;
9
10 public final class StringToDate extends UDF{
11
12     public TimestampWritable evaluate(final Text s, final Text fmt) throws Exception {
13         SimpleDateFormat sdf = new SimpleDateFormat(fmt.toString());
14         Date d = sdf.parse(s.toString());
15         TimestampWritable time = new TimestampWritable();
16         time.setTime(d.getTime());
17         return time;
18     }
19 }
```





- 一些情况下，锁和协调会是非常有用的
- 例如，如果一个用户期望锁定一个表，因为使用INSERT OVERWRITE这样的查询就可以修改表的内容，而同时第2个用户也尝试使用这个表解决某个查询问题，这样的查询可能会失败或者产生无效的结果！
- Hive可以被认为是一个胖客户端，因为在某种意义上每个Hive CLI、Thrift server或者Web接口实例都不是完全独立于其他实例的。因为这个独立性，所以锁必须由单独的系统进行协调



```
<property>
  <name>hive.support.concurrency</name>
  <value>true</value>
  <description>
    Whether Hive supports concurrency control or not.
    A ZooKeeper instance must be up and running when using zookeeper Hive lock manager
  </description>
</property>
```

```
<property>
  <name>hive.zookeeper.quorum</name>
  <value>spark001,spark002,spark003</value>
  <description>
    List of ZooKeeper servers to talk to. This is needed for:
    1. Read/write locks - when hive.lock.manager is set to
    org.apache.hadoop.hive.q1.lockmgr.zookeeper.ZooKeeperHiveLockManager,
    2. When HiveServer2 supports service discovery via Zookeeper.
    3. For delegation token storage if zookeeper store is used, if
    hive.cluster.delegation.token.store.zookeeper.connectString is not set
  </description>
</property>
```



- 配置好这些属性后，Hive会对特定的查询自动启动获取锁
- SHOW LOCKS;
- Hive提供了2种类型的锁，开启并发功能后，它们也就自动被启用了，某个表被读取的时候需要使用共享锁
- 对于其他那些会以某种方式修改表的操作都是需要使用独占锁的，不仅会冻结其他表修改操作，同时也会组织其他进行进行查询
- LOCK TABLE alibaba_log EXCLUSIVE;
- select * from alibaba_log;
- UNLOCK TABLE alibaba_log;

