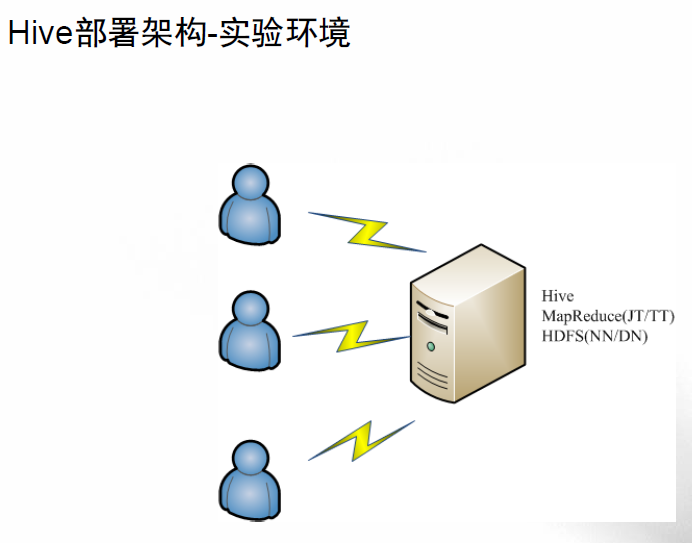
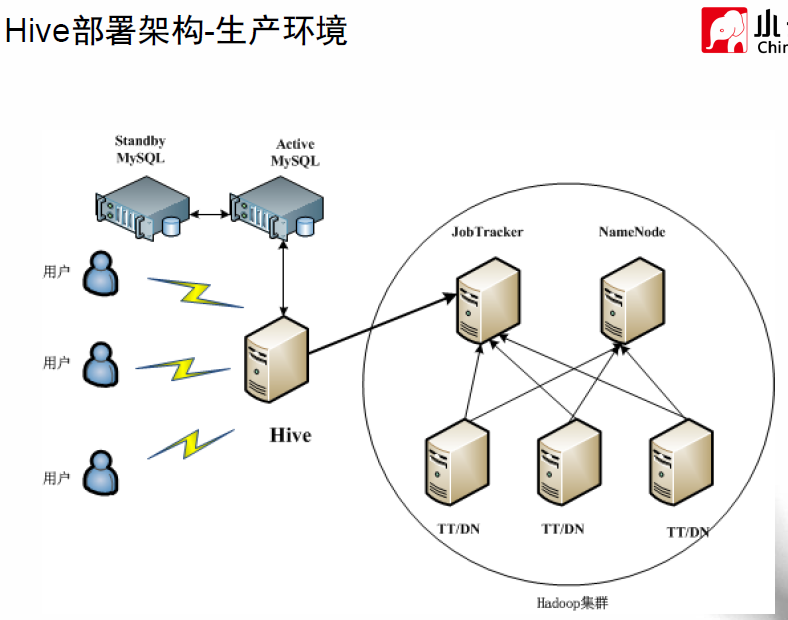
1. Hive背景及应用场景。
   1. Hive是什么？
      1. 由Facebook开源，最初用于解决还量化结构化的日志统计问题。
         1. ETL（Extraction-Transformation-Loading）工具
      2. 构建在hadoop之上的数据仓库；
         1. 数据计算使用MR，数据存储使用HDFS
      3. Hive定义了一种类SQL查询语言-HQL
         1. 类似于SQL，但不完全相同
      4. 通常用于离线数据处理（采用MapReduce）
      5. 可认为是一个HQL-MR的语言翻译器。
2. Hive典型应用场景
   1. 日志分析
      1. 统计网站一个时间段内的pv、uv
      2. 多维度数据分析
      3. 大部分互联网公司使用Hive进行日志分析，包括百度、淘宝等。
   2. 其他场景
      1. 海量结构化数据离线分析
      2. 低成本进行数据分析（不要直接编写MR）
3. 为什么使用hive？
   1. 简单，容易上手
      1. 提供了类似SQL查询语言HQL
   2. 为超大数据集设计的计算/扩展能力
      1. MR作为计算引擎，HDFS作为存储系统。
   3. 统一的元数据管理（HCalalog）
      1. 可与Pig、Presto等共享
4. 有了Hive，还需要自己写MR程序吗？
   1. Hive的HQL表达的能力有限
      1. 迭代式算法无法表达
      2. 有些复杂运算用HQL不以表达
   2. Hive效率较低
      1. Hive自动生成MapReduce作业，通常不够智能。
      2. HQL调优困难，粒度较粗。
      3. 可控性差。
5. Hive基本架构

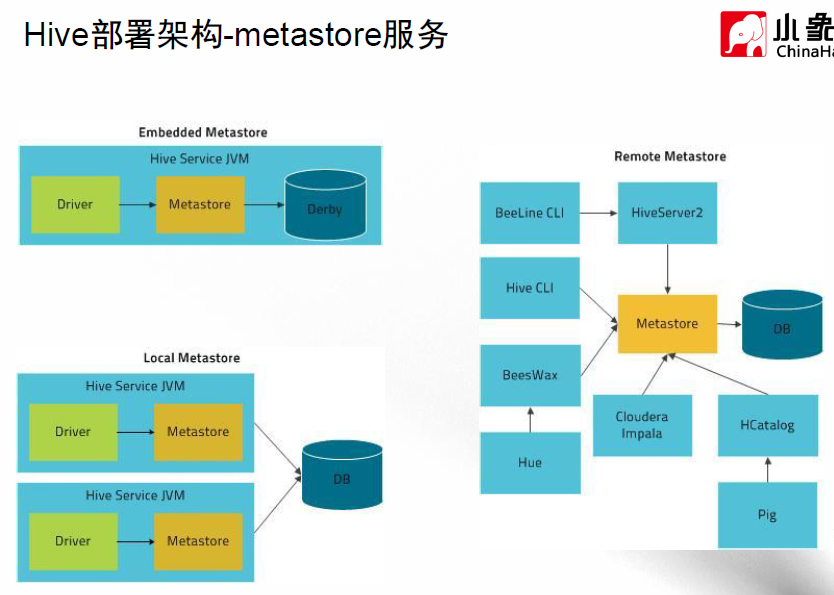


Hive各模块组成

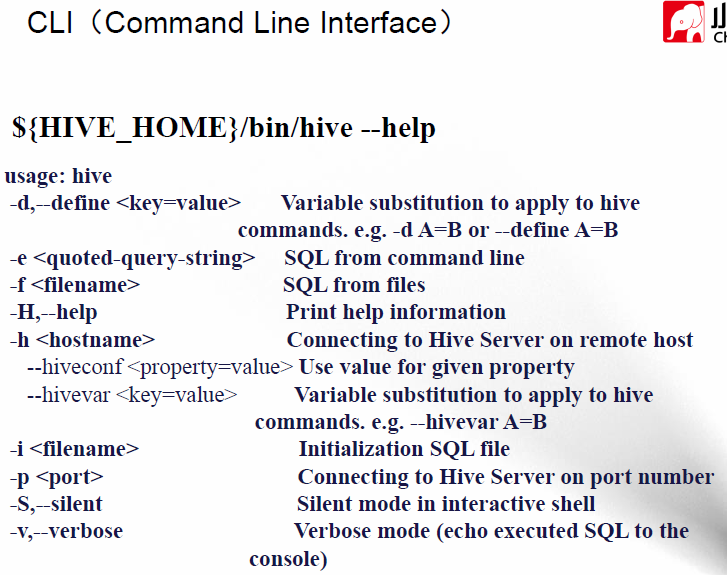
1. 用户接口
   1. 包括CLI、JDBC/ODBC、WebUI
2. 元数据存储（metastore）
   1. 默认存储在自带的数据库derby中，线上使用时一般换为MySQL。
3. 驱动器（Driver）
   1. 解析器、编译器、优化器、执行器
4. Hadoop
   1. 用MapReduce进行计算，用HDFS进行存储。



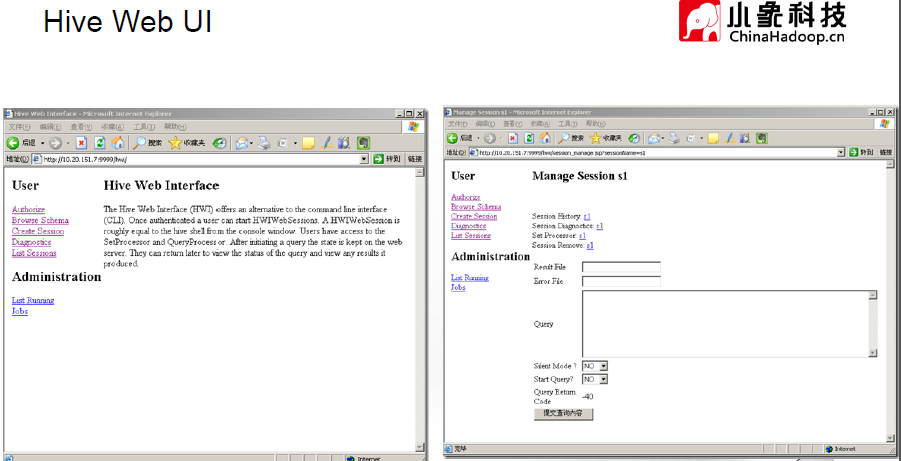




Hive的使用方式：

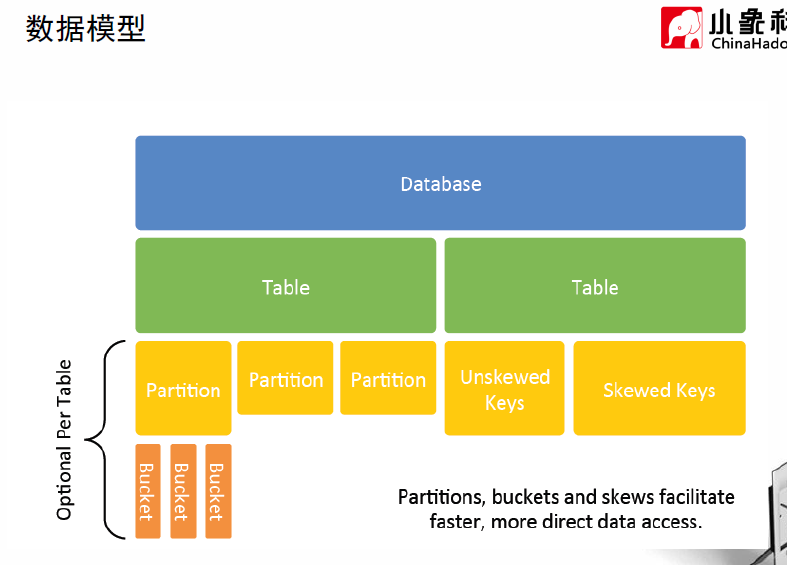
1. 
2. CLI-hive外部资源
   1. 外部资源：HQL运行时需要的jar包、二进制文件、文本文件、压缩文件等。
   2. 外部资源需发到集群的各个节点上使用；
   3. 三种外部资源：
      1. FILE：普通文件，hadoop不会进行任何处理
      2. JAR：jar包，hadoop自动将其加入CLASSPATH中
      3. ARCHIVE：归档文件，Hadoop可识别“.tgz”、“.tar.gz”、“.zip”等结尾的文件，并自动解压





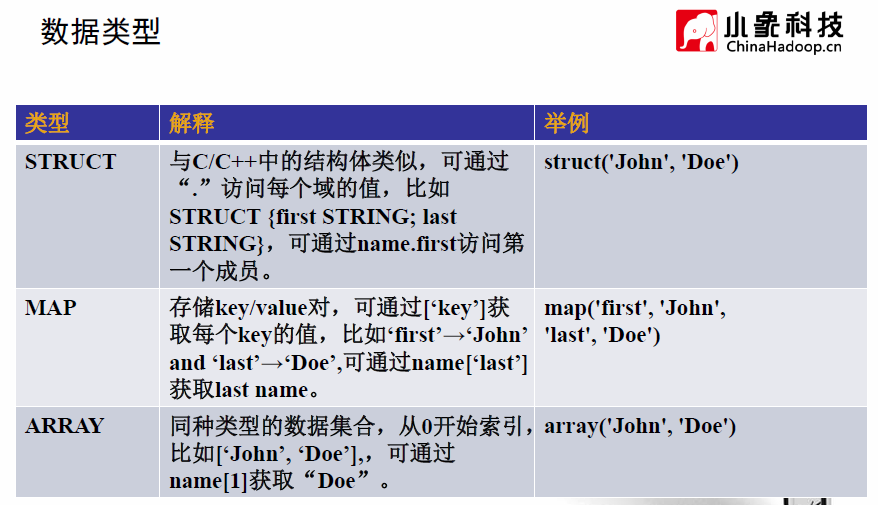


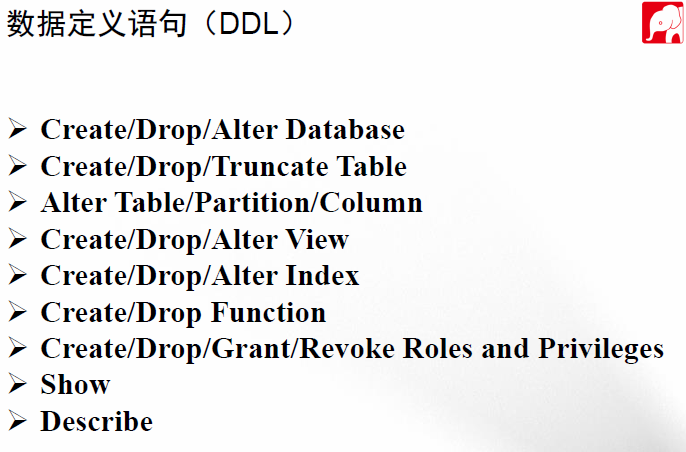
数据模型：



桶：类似与索引。







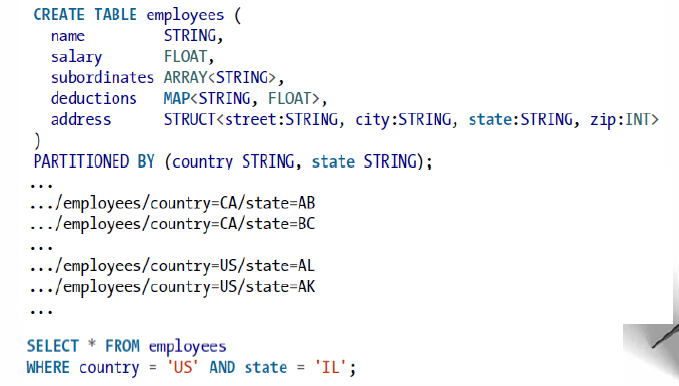




Hive Partition与Bucket

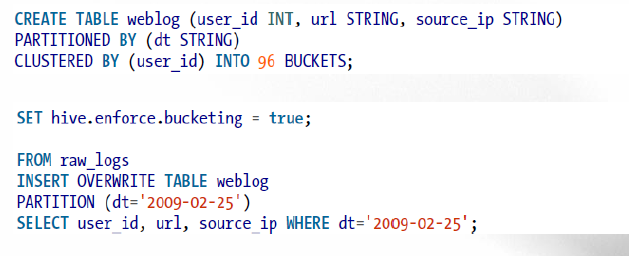
Partition：

1. 为了减少不必要的暴力扫描，可以对表进行分区
2. 为避免产生过多的小文件，建议只对离散字段进行分区



Bucket：

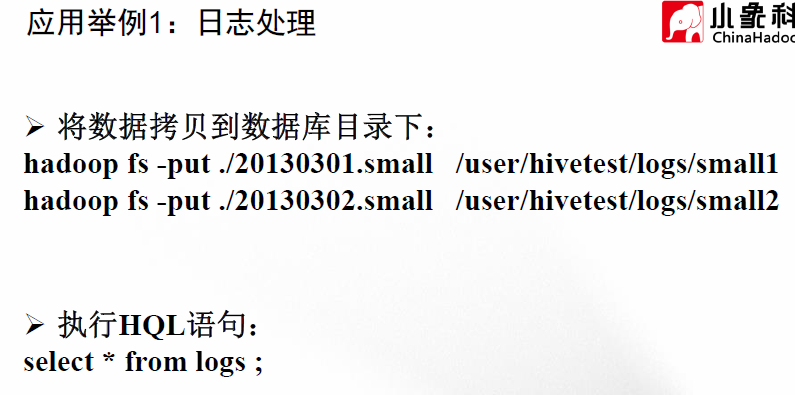
1. 对于值较多的字段，可将其分成若干个Bucket
2. 可结合Partition与Bucket使用（Bucket：适合离散的数据的字段）



Hive数据格式：

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



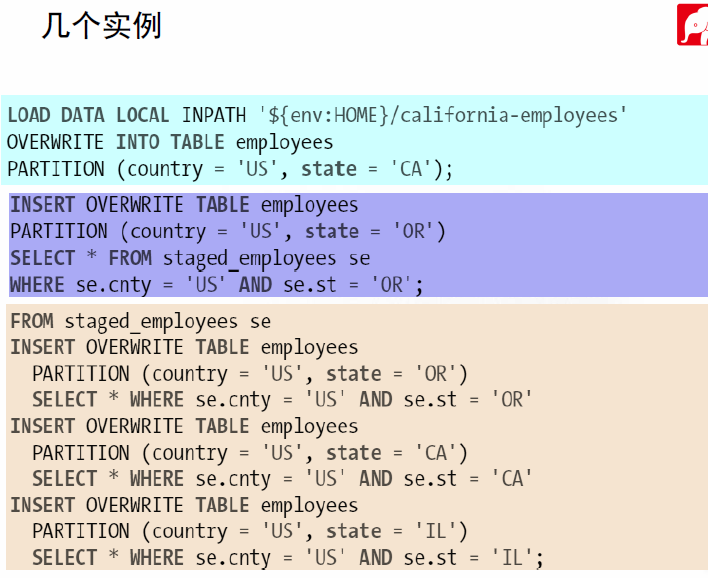
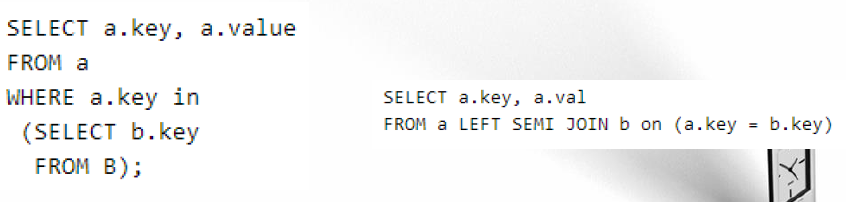
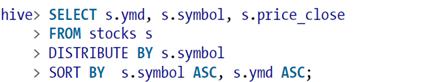
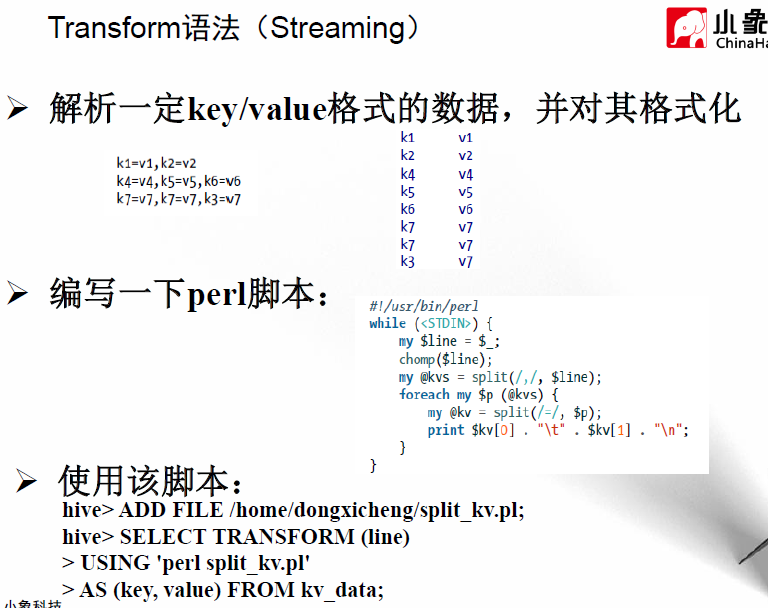
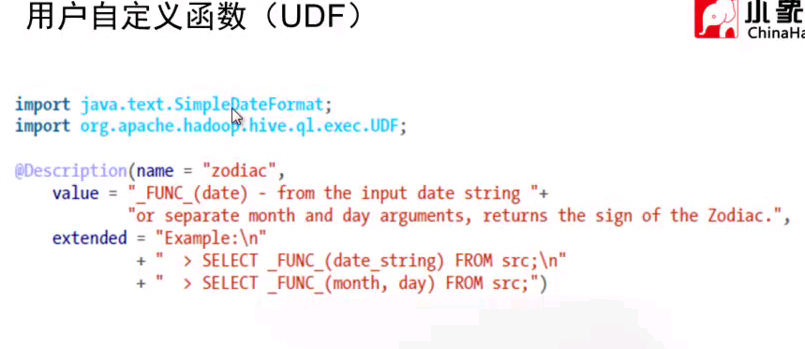
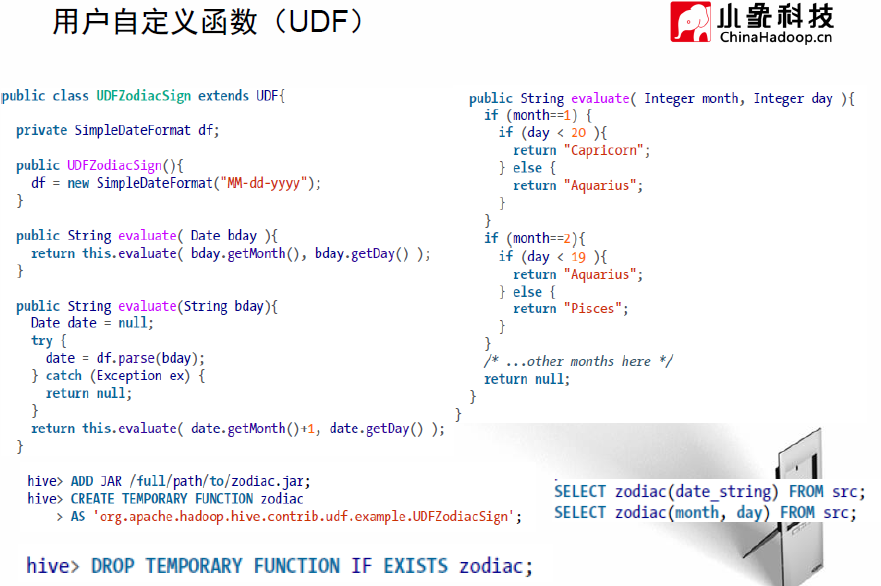
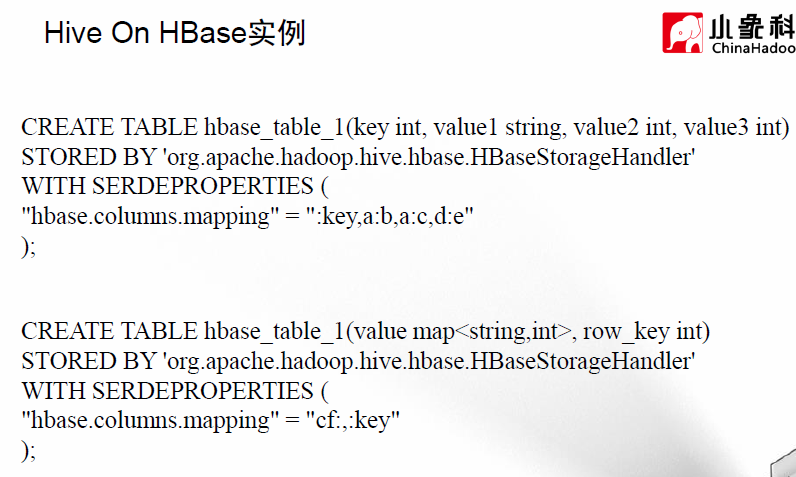


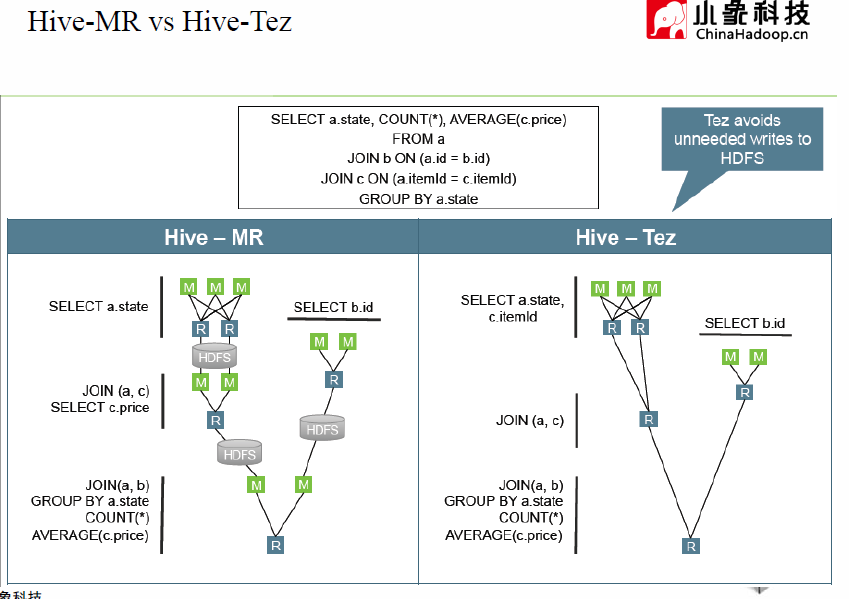
数据操作语句（DML）

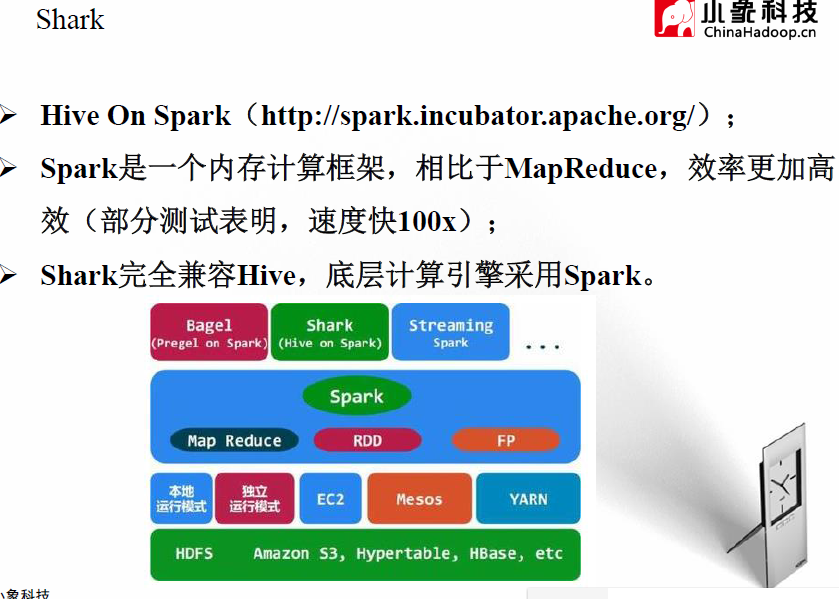
1. 数据加载与插入语句：
   1. LOAD
   2. INSERT
2. 数据查询语句
   1. SELECT
3. 查看HQL执行计划
   1. Explain
4. 表/分区导入导出
   1. Export/Import



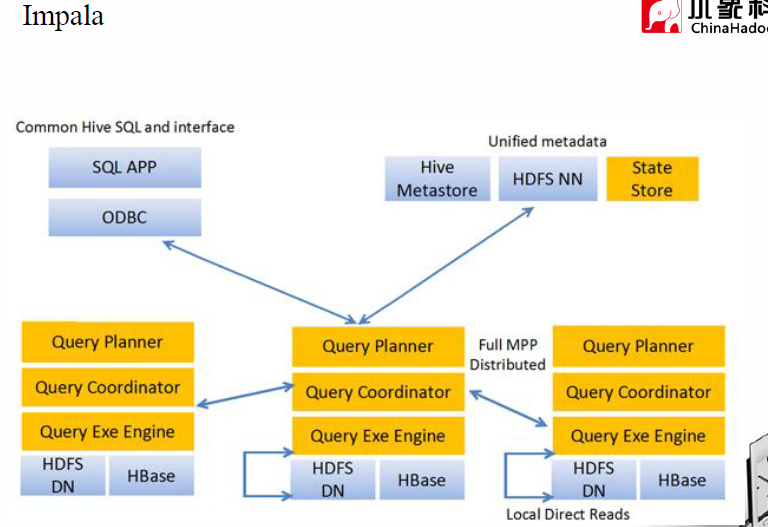
数据加载与插入语句：

1. Load data
   1. 当数据被夹在至表中时，不会对数据进行任何转换。Load操作知识将数据复制/移动至Hive表对应的位置
   2. 默认每个表是一个目录，比如数据库dbtest中，表名为tbtest，则数据存放位置为：
      1. ${metastore.warehouse.dir}/dbtest.db/tbtest
   3. Metastore.warehouse.dir默认值是
      1. /usr/hive/warehouse
2. 
3. 
4. 
5. 两种分布式Join算法
   1. Map-side join（Broadcat Join）
      1. Join操作在Map Task中晚餐，因此无需启动reduce task；
      2. 适合一个达标，一个小表的连接操作；
      3. 思想：小表度知道各个节点上，并加载到内存中；达标分片，与小表晚餐连接操作；
   2. Reduce-side Join(Shuffle join)
      1. Join操作在Reduce Task中晚餐；
      2. 适合两个达标连接操作；
      3. 思想：map端按照连接字段进行hash，reduce端完成连接操作。
6. HQL中两种特殊Join
   1. Map-side join（broadcast join）
      1. SELECT /\*+ MAPJOIN(b) \*/ a.key, a.value FROM a join b on a.key = b.key
   2. LEFT SEMI JOIN（左半连接）
      1. 
   3. Order By与Sort By
      1. Order By
         1. 启动一个Reduce Task
         2. 数据全局有序
         3. 速度可能会非常慢
         4. Strict模式下，必须与Limite连用
      2. Sort By
         1. 可以有多个reduce task
         2. 每个Reduce Task内部数据有序，但全局无序
         3. 通常与distribute by
   4. Distribute by与Cluster by
      1. Distribute by
         1. 相当于MapReduce中的Partitioner，默认是基于hash实现的；与sort by连用，可发挥很好的作用
         2. 举例：
            1. 
      2. cluster by
         1. 相当于distribute by与sort by(降序)连用，且跟随的字段相同，可使用cluster by简写；
         2. 
   5. Transform语法（Streaming）
      1. Hive允许使用任何语言编写Mapper和Reducer并嵌套到HQL中使用
      2. 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。
      3. 使用操作符TRANSFORM
      4. 
   6. 用户自定义函数
      1. UDF：扩展HQL能力的一种方式
      2. 三种UDF：
         1. 普通UDF（1对1）
         2. UDAF：用户自定义聚集函数（多对一）
         3. UDTF：用户自定义产生函数（1对多）
      3. 函数相关操作：
         1. SHOW FUNCTIONS；
         2. DESCRIBE FUNCTION concat；
         3. DESCRIBE FUNCTION EXTENDED concat;
         4. SELECT concat(column1,column2) AS x FROM table;
      4. 
      5. 
      6. 
      7. 其他问题
         1. HQL支持索引吗？
            1. HQL执行过程主要进行暴力扫描。莫倩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；
         2. Hive支持update操作么？
            1. 不支持，hive底层是HDFS，HDFS仅支持追加，不支持随机写。
         3. Skew Data处理随机机制？
            1. 指定skew列：CREATE TABLE list\_bucket\_single (key STRING, value STRING) SKEWED BY (key) ON (1,5,6);
            2. 为skew task分配更多资源（TODO）
            3. 将skew分解成多个task，再合并结果（TODO）
      8. Hive On Hbase
         1. 使用HQL处理Hbase中的数据。
            1. 比直接通过Hbase API存取数据方便；
            2. 但是性能更低，相当于把在线处理转化为批处理。
         2. 存在问题
            1. 不够成熟
            2. 不能按时间戳获取数据，默认总是存取最新的数据。
         3. 
      9. Stinger：
         1. 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；
         2. 下一代被称为“Stinger”，其底层的计算引擎将由Tez替换MapReduce；
         3. Tez相比于MapReduce具有众多优势：
            1. 提供了多种算子（如MapReduce、Shuffle）供用户使用；
            2. 将多个作业合并成一个作业，减少磁盘读写IO；
            3. 充分利用内存资源。





Impala：

1. 底层计算引擎不再采用MR，而是使用商用关系数据库类似的分布式查询引擎；
2. Impala可直接处理HDFS上的数据，并将结果集合再次写入HDFS；
3. 具有良好的扩展性和容错性；
4. 适合快速交互式查询
5. 
6. 