**Q1: 我做完算子运算以后进行count操作显示成功，但是在落盘的时候出现内存异常，分析看是在分布式计算的时候出现问题。能否帮忙分析一下count的过程，以及后续是否可以继续使用count进行逻辑验证，如果没有，是否有其他的方式。**

***老师：***Spark的计算是lazy execution，只有action的算子才会触发真正的计算，count就是其中一个会触发真正的计算。但是，这些action的算子，比如write，会直接写入最终的目的地。如果目的地是HDFS的话，所有的executor的task都会直接并行写入HDFS，但是count不一样，count最后的结果是收集在driver里的。所以出现的内存异常一般来说，都是driver的内存溢出，这个时候建议增大driver的内存。另外，如果想做逻辑验证也可以使用spark的dataframe的show来进行逻辑验证，可以只show前20行数据，这样driver就不会内存溢出了，因为show的结果也是最后发送到driver上的。一个count的操作的顺序就是：

输入地址（HDFS）---》读取到各个executor中的task中，然后每个task进行count，最后汇总到driver上。同时，count还会根据你输入地址里存储的文件格式不同，会有不同的操作。如果输入文件是text文本文件，那么对读取整个文件，然后进行count计数。如果是parquet或者orc这样的列式存储文件，那么这些列式存储文件里会存有一定的统计数据，不会去读取整个文件。

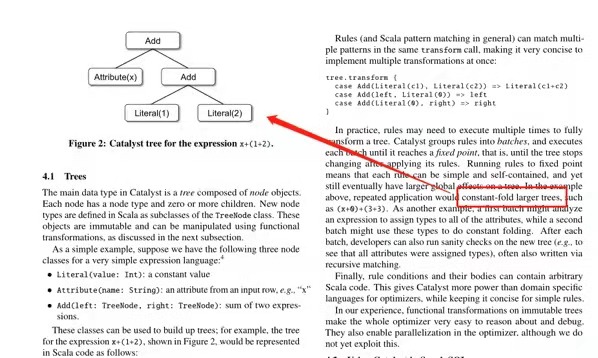
***追问：***count应该不会消耗太多driver的内存吧？比毕竟只保存一个计数？

***老师：***保存结果是不会消耗太多的内存，但是最后汇总count的数据的时候，会消耗对应的driver内存。也就是最后reduce的时候，都是reduce到一个driver，driver要从各个executor的task中拉取结果。

***追问：***那应该也不会消耗太多吧？拉取的应该是每个executor上每个分区的计数吧？

老师：是的，但是如果说数据量很大，executor开了很多task，那么在拉取的时候，就会占用driver的内存，而driver拉取的时候是要拉取所有分区的计数的。如果没有拉取完所有的分区的技术结果的时候，driver的内存是不会释放的。

**Q2: 请问一下这个恒定倍数折叠大树是怎么理解呢**

****

***老师：***首先，这个是Spark的Catalyst中出现的，Catalyst是Spark sql的优化器，是来优化用户写的SQL语句。举个简单的例子，如果用户的sql语句在前后会遍历两次同一张表，那么这个优化器会优化成让用户只遍历一个表。而这个优化器的实现主要是通过树的形式进行数据的存储和计算的。Scala中新定义的node类型是TreeNode类的子类。这些对象都是不可改变的，可以使用函数转换来操作，而Spark就是Scala编写的。而图中的给出的例子中，所形成的树，是有三种类型的node，

1. Literal(value: Int): 一个常量

2. Attribute(x):输入变量 “x”

3. Add(left: TreeNode, right: TreeNode):两个expressions求加

按照图中的x+(1+2)的形式构建对应的树，然后进行优化

构建完这些树后，我们会有不同的规则来进行树的优化。

举个例子x+ (y+1)+(y+2)+ (4+5)

这个表达式expression我们会先会构建最初始的树来表示它，然后利用Spark的catalyst中定义的规则进行优化。 最后可以优化成 x +2y + (1+2+4+5)的新树，那么最后的优化结果就用这个新树来表示。

这样的优化，可以避免用户去重复地遍历同一个数据，减少加载到内存中的数据量，精简计算过程，从而让Spark计算更加高效。

***追问：***但是还是不太理解那个constant folding tree是怎么样优化的

***老师：***是根据不同的rule来进行优化的，就是说，当有一个初始树A树，然后根据对应的rule进行了优化后会形成最后的B树结果，rule的讲解应该在你发的截图的下一节。对于rule的具体优化，会涉及到数学，如果有需要，我可以录个20多分钟的视频，给大家讲解。

**Q3: 请问一下这个问题该怎么回答**

***老师：***真正意义上cluster traversal是只有一次，因为我前面说过Spark的计算都是lazy execution，只有真正的action算子，才会触发计算过程，数据才会在集群中进行遍历和计算。但是有关SQL的优化，以及逻辑和物理的执行顺序是在driver里面生成的，然后被count的算子触发进行计算。

***追问：***但是第一次读取textfile也算cluster traversal吧，这样就两次了啊

***老师：***读取textfile的数据的时候其实只有一次，刚开始再进行执行顺序的生成过程中，其实只是读取了hdfs中的相关元数据，但是对于数据本身，并没有直接的加载。

***追问：***只有action算子才会触发且只会触发一次cluster traversal吗？有计算过程中有shuffle的情况呢？但是lines的生成不需要cluster traversal吗

老师：只有action算子才会触发计算，才会载入真正的数据，如果没有action的算子，也不会存在shuffle的情况发生。

***追问：***有action且存在shuffle的情况会有多次cluster traversal吗？

***老师：***需要的，但是真实的情况是，当count触发了这一次的数据载入后，比如指定了100个task，那么这100个task中的每一个task就会做，read data，filter,然后count，一气呵成，并没有说是，会把中间结果存放在哪里。是的，会有多次的cluster traversal，而Spark中对于stage的划分就是根据shuffle或者说cluster traversal来划分的

***追问：***所以是count触发之后，才会去read在hdfs里的file吗

***老师：***是的，去read hdfs中的真正数据

<https://github.com/apache/spark/blob/master/sql/catalyst/src/main/scala/org/apache/spark/sql/catalyst/expressions/aggregate/Count.scala#L26>

这个是count的source code，大家可以看到 Returns the total number of retrieved rows, including rows containing null最后return的就是每个task处理的数据量。然后在driver里进行聚合

**Q4: 直接安装pip install pyspark 和按正确方式安装的区别**

***老师：***Pip install是直接在线从pypi网上下载，而从网上下载下来再安装，是正常安装软件和框架的方式。

有一个不同是，pip install一般不会帮你去配置SPARK\_HOME的路劲，所以如果下载完后找不到pyspark，需要自己去进行设置。同时，pip install后的pyspark是直接默认本地部署的，一般来说它支持所有的pyspark的function，但是有极个别的可能需要更新。因为，pyspark的更新速度是要比pip里面的pyspark要快一些。有关这个问题，大家可以查看pypi的网站https://pypi.org/project/pyspark/#history

***追问：***那底层有hive hadoop的东西么

***老师：***hive hadoop，并不是spark的package里面，会有spark jar，但是有关hive和hadoop需要再download，同时如果想部署在hadoop yarn上，也需要再download。hadoop yarn

***追问：***如何配置hive使用spark引擎？老师之后能出视频讲解一下吗？

***老师：***好的，没有问题。

**作业1：现在思考一些问题，来看一下大家是否真的理解**

****

老师：这个问题，首先贴一个同学的答案，基本上是正确的。但是，有一些问题，大家可以看一下，哪里有点小问题。

sql语句发送到hive server2的client端

1.编译器Compiler根据用户任务去访问matestore服务，从数据库(mysql)获取元数据信息，然后对任务进行编译。

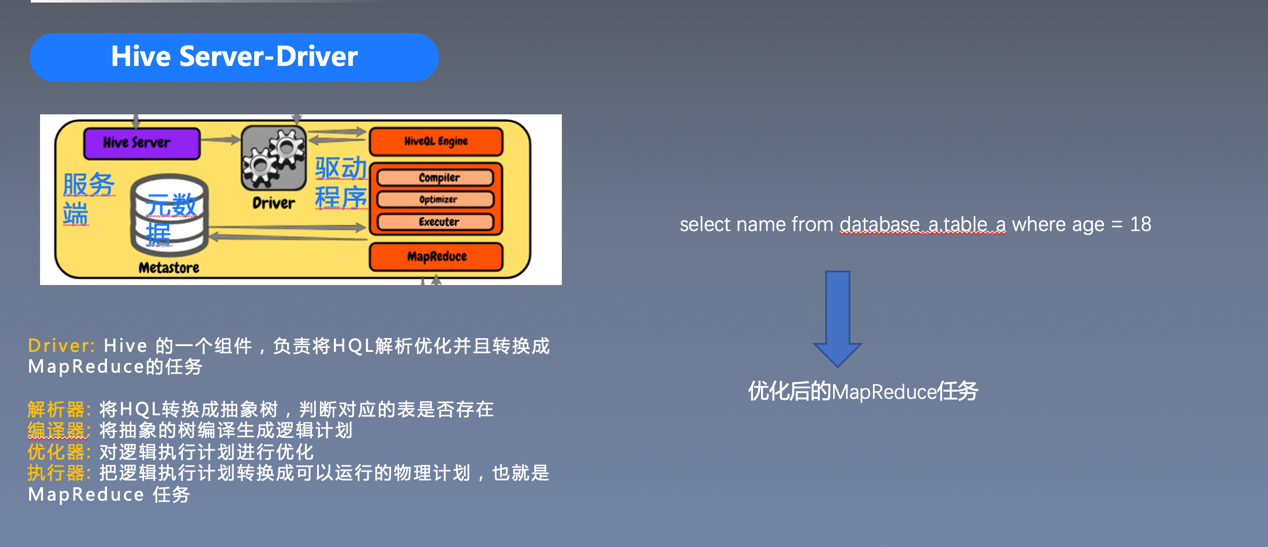
2.通过解析器Parser将sql解析为AST（抽象语法树），会进行语法校验。

3.分析器Analyzer进行语法解析，生成语句块

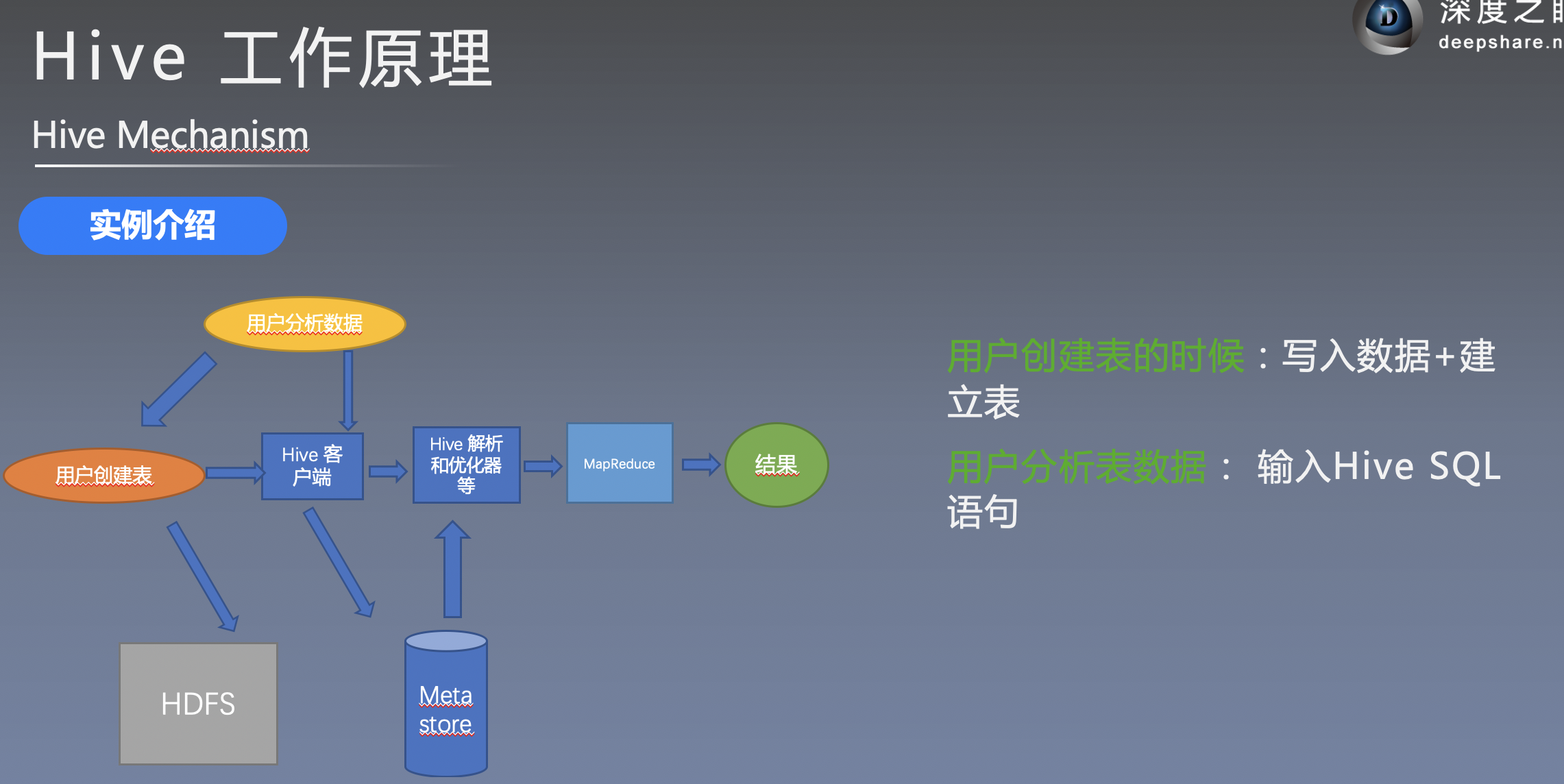
4.编译器Compiler将逻辑执行计划解析，生成一堆Opertator Tree

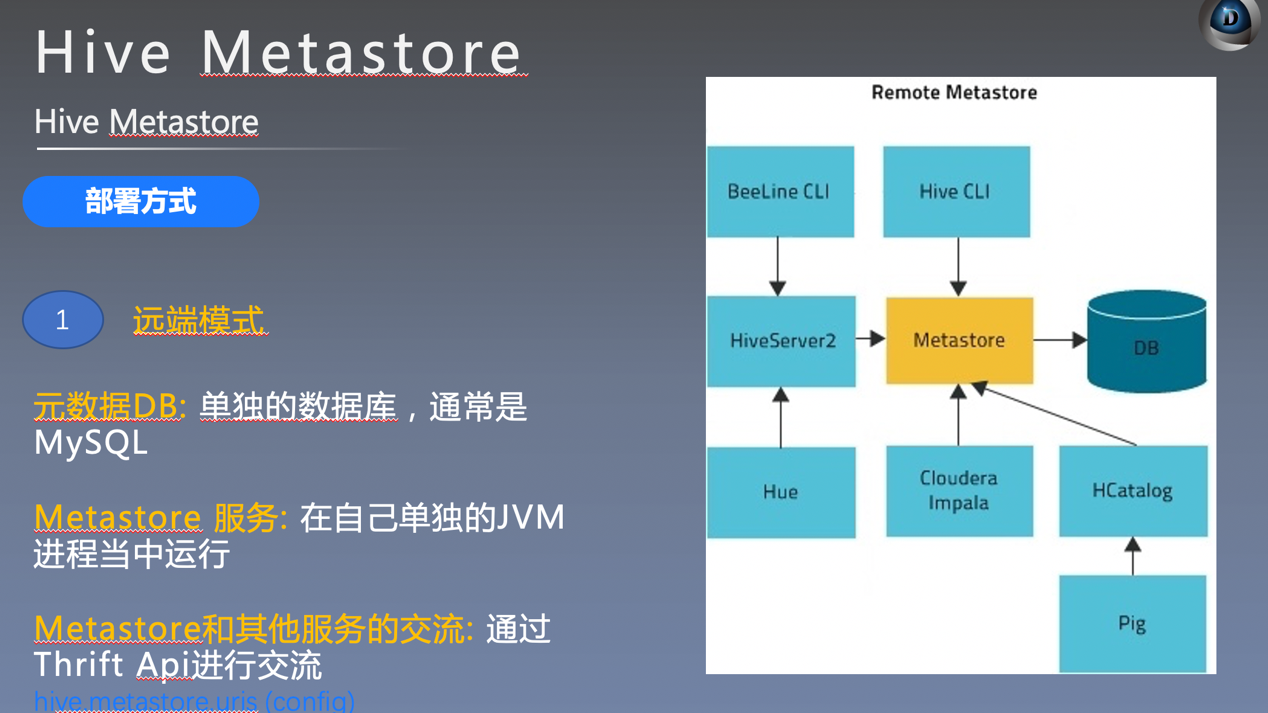
5.优化器Logical optimizer:进行逻辑执行计划优化，生成一堆优化后的Opertator Tree

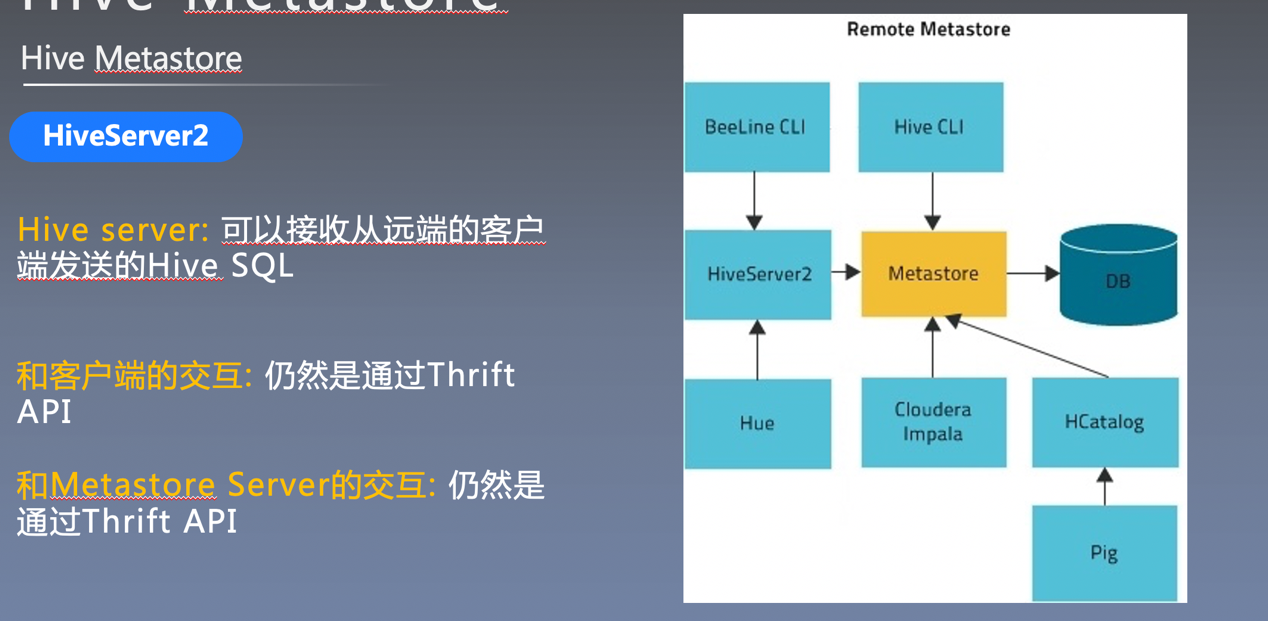
6.物理执行计划解析，生成tasktree，并优化，然后再提交到集群上面去

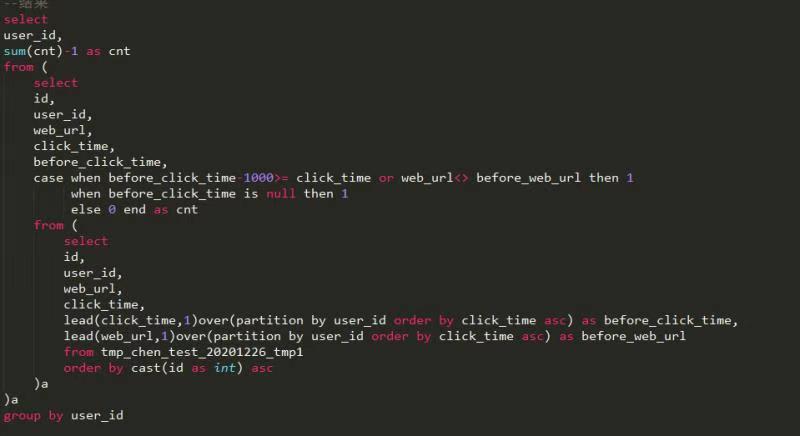


结合这几个图，大家可以看一下









这个是这个同学的解法，也发送给大家，我也再发送一个其他的解法给大家

有关这里，正确的步骤应该是sql语句从Hiver server 2的client端进行提交，然后提交到hive server端。

1.编译器Compiler根据用户任务去访问metastore服务，

2. metastore服务会从元数据库(mysql)获取元数据信息，然后对任务进行编译。

3.解析器Parser将sql解析为AST（抽象语法树），会进行语法校验。

4.分析器Analyzer进行语法解析，生成语句块

5.编译器Compiler将逻辑执行计划解析，生成一堆Opertator Tree

6.优化器Logical optimizer:进行逻辑执行计划优化，生成一堆优化后的Opertator Tree

7.物理执行计划解析，生成tasktree，然后再提交到集群上面去

**作业2：Hive SQL面试考题，P7级**

****

老师：有关第二个sql语句的问题，大家可以也看一下不用case when的解法，稍晚点也会直接发送到群里。有关这次答疑中提到的rule的问题，我会录制视频讲解。对于进度，也会加快，希望能巩固大家的记忆。

**互动环节：Spark on hive 与 Hive on Spark 的区别是什么？**

spark on hive指的是用spark去支持hive的语句，大家可以用hive的语法直接在spark里进行编写。比如hive语句是 select \* from table where year = 2020，那么在spark里就可以直接利用spark.sql（“select \* from table where year = 2020”）这样的代码方式，进行hive语句的计算。但是最后执行的逻辑仍然是转换成rdd的形式。

hive on spark是把hive的map reduce 变成spark的执行引擎，需要对hive进行配置，必须重新编译和导入spark的package

但是，因为spark on hive会完全支持基本上所有的hive语句，所以之后用spark on hive就可以了。