## 2021-2-10 年29

### 体验腾讯云上的clickhouse

2个节点的集群：



另外购买一台广州三区的CVM，通过内网方式访问这个clickhouse集群。

如果需要分布式的库表：

1. 在**每个节点**都创建相同的本地表，表引擎指定为MergeTree
2. 在**每个节点**都创建Distributed表，它本身不存储数据，可以认为是多个本地表的merge或者说视图，有点像mysql的merge表

本地表：

CREATE TABLE tutorial.hits\_v1

(

`WatchID` UInt64,

`JavaEnable` UInt8,

…

)

ENGINE = MergeTree()

PARTITION BY toYYYYMM(EventDate)

ORDER BY (CounterID, EventDate, intHash32(UserID))

SAMPLE BY intHash32(UserID)

分布式表：

CREATE TABLE tutorial.hits\_v1\_all

(

`WatchID` UInt64,

`JavaEnable` UInt8,

…

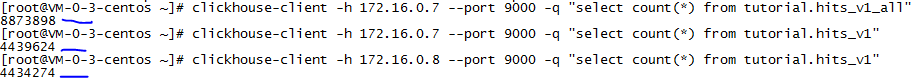
)

ENGINE = Distributed(default\_cluster, tutorial, hits\_v1, WatchID)

Distributed()函数的参数依次为：

集群名字、database名、本地表名字、sharding的字段。 Sharding也可以是rand（）等方式。

可以看到下面的三个表的行数是有加和关系的：



### 体验superset

Superset是apache的一个顶级开源数据可视化项目，类似grafana，用于数据图表呈现。

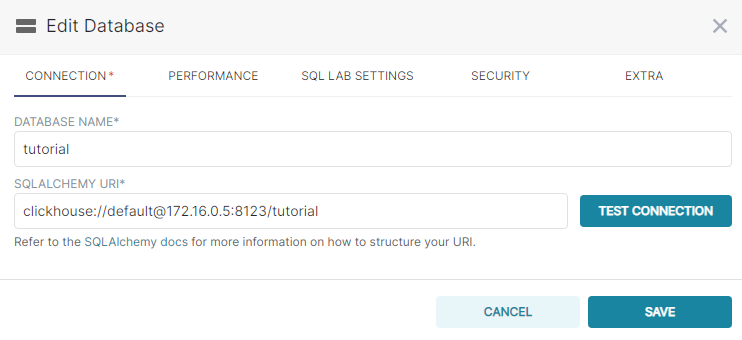
Docker镜像仓库里有好几个superset的镜像，

其中star数最多的docker.io/amancevice/superset是靠谱的，那个看似官方的apache/superset挺不靠谱的，没有考虑通过volumn持久化的问题，很初级。

按照<https://github.com/amancevice/docker-superset>的指引，把/etc/superset和/var/lib/superset挂载到宿主机的空目录（所有人有读写权限，否则报错），然后执行superset-init初始化数据，就可以跑起来了。

docker run -d --name superset -p 8088:8088 -v /etc/localtime:/etc/localtime:ro -v /etc/superset:/etc/superset:rw -v /var/lib/superset:/var/lib/superset:rw amancevice/superset

然后打开页面就可以添加前面的clickhouse这个数据源了：



clickhouse://default@172.16.0.5:8123/tutorial

因为是没有密码的，所以是上述URI，有网文写说用”click”密码，即

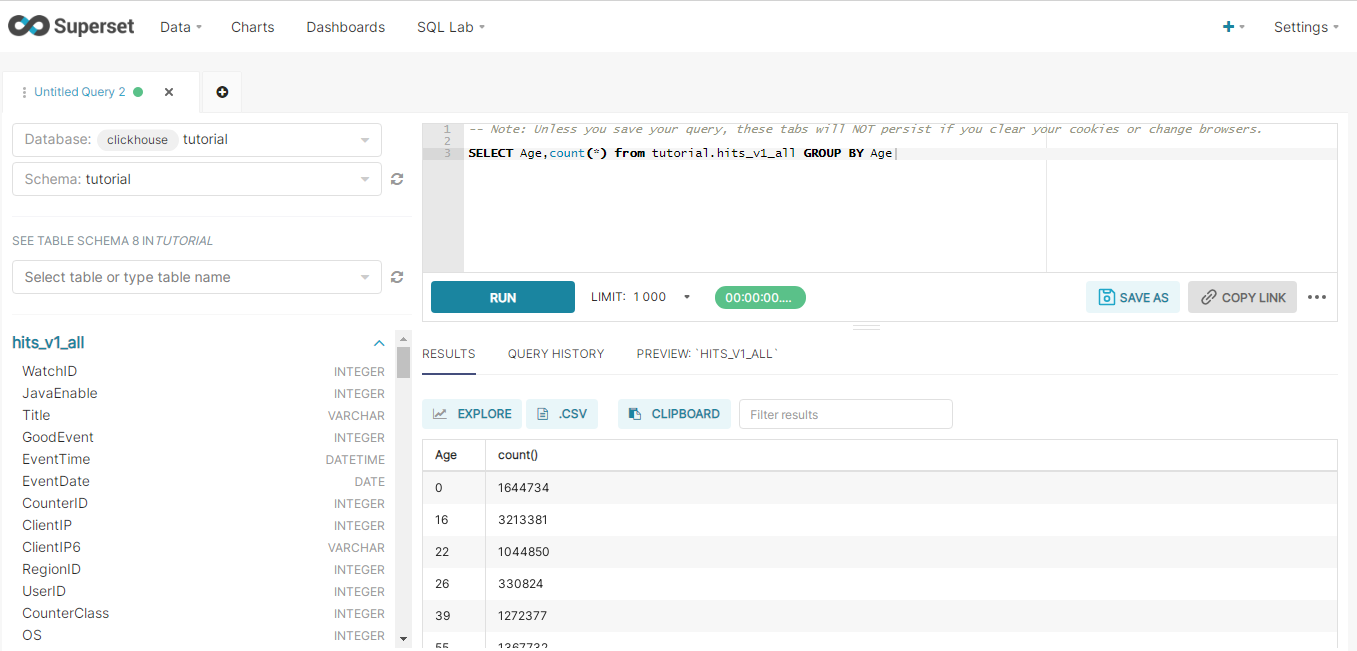
clickhouse://default:click@172.16.0.5:8123/tutorial，这个连不上。

只要是支持用python的SQLALCHEMY方式访问的数据库，都可以添加为superset的数据源。

Superset的功能，与grafana的visulize/dashboard很类似，例如统计hits表里年龄的分布：



Chrome浏览器下，superset的SQL Lab界面有些问题，换MS Edge浏览器就ok了：



## 2021/2/11 年30

### mahout

简单看了一下介绍，mahout是apache的一个顶级项目，主要是基于hadoop实现机器学习算法的在集群上并行化/分布式处理，也就是算法的M-R化，包括的算法有：

1. 线性代数算法
   1. 矩阵的QR分解
   2. 主成分分析PCA
   3. 奇异值分解SVD
2. 数据的预处理
   1. One-hot编码
   2. 均值中心化处理
   3. 归一化+均值中心化处理
3. 回归算法
4. 聚类算法
5. 推荐算法

### mllib

spark体系下的机器学习库，直接拷贝官网上的文字，看起来比mahout支持的更丰富：

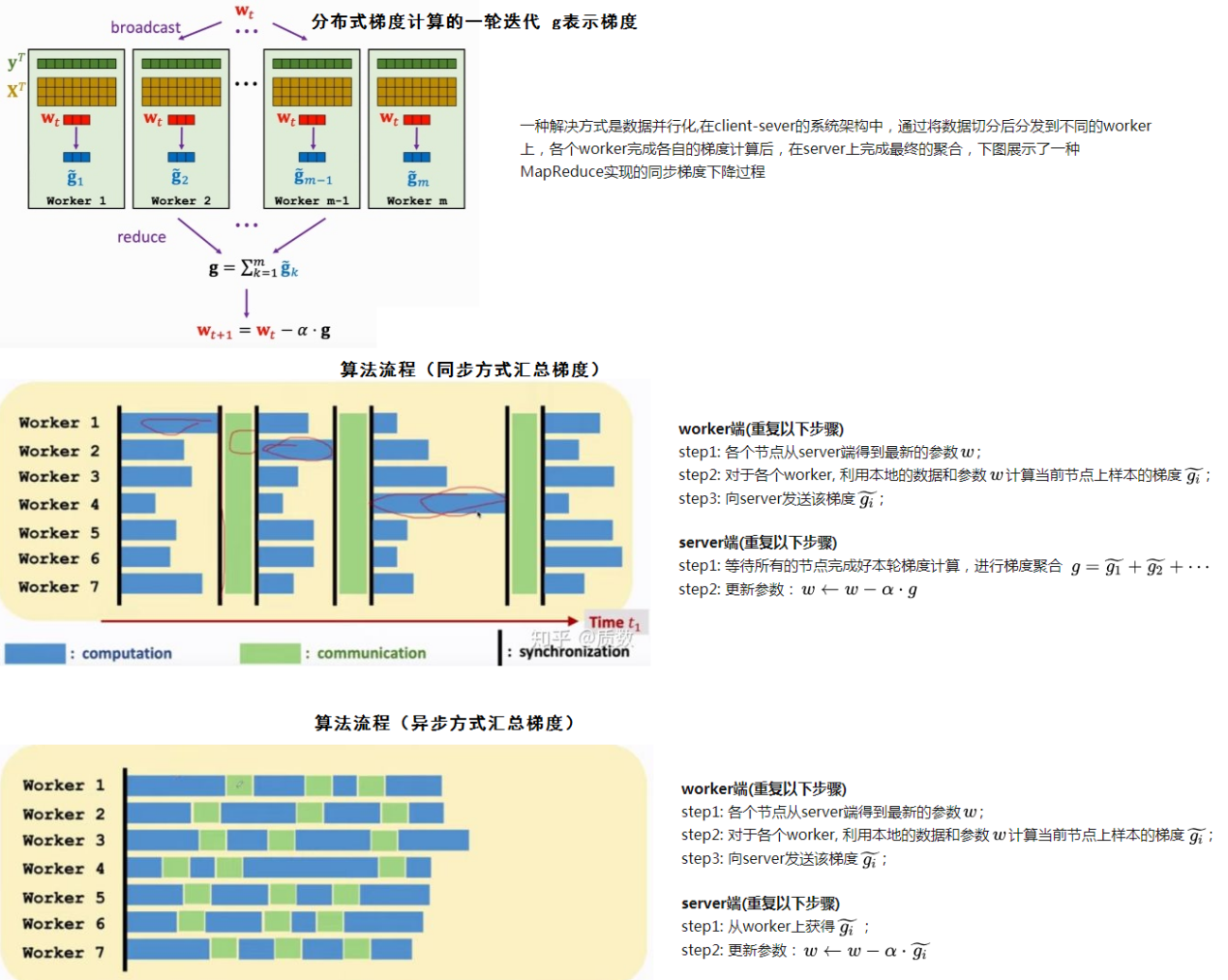
ML algorithms include:

* Classification: logistic regression, naive Bayes,...
* Regression: generalized linear regression, survival regression,...
* Decision trees, random forests, and gradient-boosted trees
* Recommendation: alternating least squares (ALS)
* Clustering: K-means, Gaussian mixtures (GMMs),...
* Topic modeling: latent Dirichlet allocation (LDA)
* Frequent itemsets, association rules, and sequential pattern mining
* Distributed linear algebra: SVD, PCA,...
* Statistics: summary statistics, hypothesis testing,...

机器学习的库很多，可能mllib、mahout的主要特点是分布式实现，类似深度学习里吵得很火的联邦学习。

### 使用梯度下降的机器学习算法的分布式实现

知乎网友：联邦学习（Federated Learning）实际上是一种加密的分布式机器学习技术，参与各方可以在不披露底层数据和底层数据的加密（混淆）形态的前提下共建模型。它可以实现各个企业的自有数据不出本地，而是通过加密机制下的参数交换方式，即在不违反数据隐私法规的情况下，建立一个虚拟的共有模型。在这样一个机制下，参与各方的身份和地位相同，成功实现了“共同富裕”的目标。



不使用梯度下降的算法，也是可以分布式并行化的，例如k-means聚类算法，就有map-reduce的实现方式。

只要样本间不发生太多交叉运算，只计算单个样本的某种值（例如与质心的距离），那么该算法就通常可以实现分布式并行化。

## 2021-2-12 年初一

### HIVE

简单地说，hive就是用sql的接口访问map-reduce。

看文档就觉得胆颤心惊，感觉我搞不定手工搭建环境，老老实实在腾讯云买了一套HIVE+Hadoop环境。

腾讯云EMR在服务器上的目录规则：

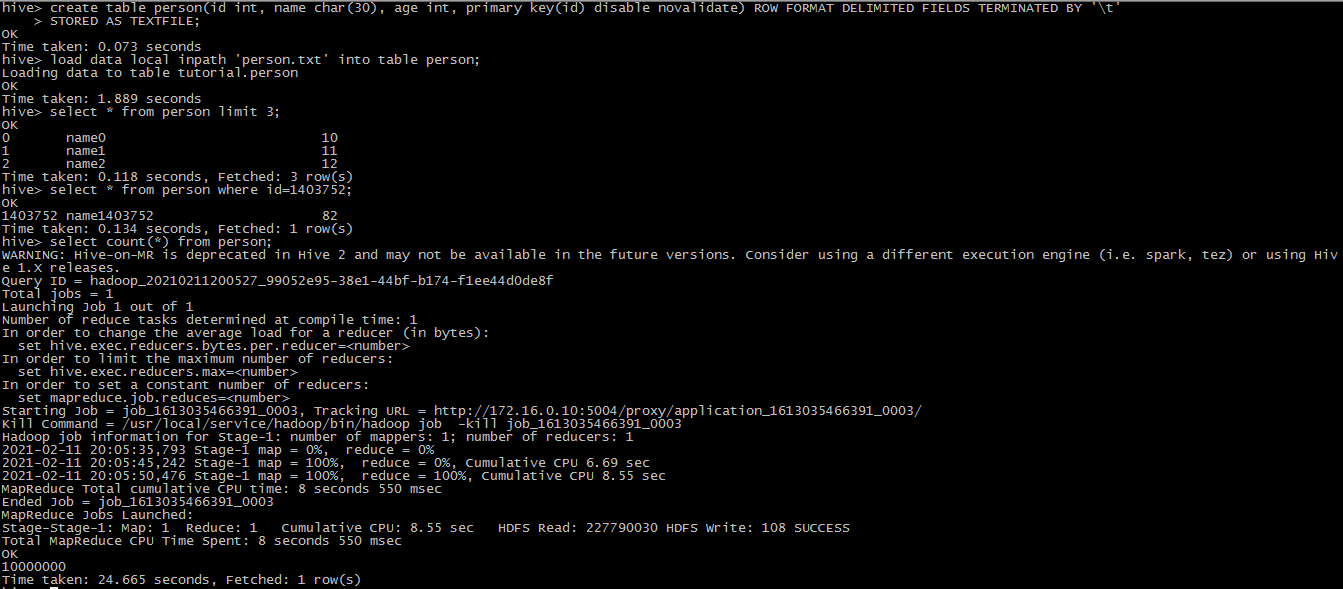
1、/usr/local/service/下是各个组件的安装目录

2、/data/emr下是各个组件的数据和日志等

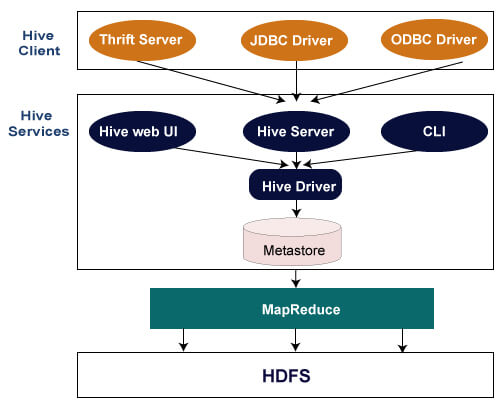
遇到几个小坑：

1. HIVE Cli的删除键不起作用，修改SecureCRT的终端仿真为linux就好了。
2. Root用户打开hive Cli，会提示没有权限创建表。su到hadoop用户就可以了

如文档所说，hive通过map-reduce方式并发访问底层数据，时延大，不适合OLTP：



Hive的架构:



Hive的精髓是一种灵活的外部表，通过指定文件位置、文件格式，就可以基于文件构建一张表：

1. 创建外部表：  
   hive> create external table goods(id int, name char(30), price int) row format delimited fields terminated by ',' location '/HiveDirectory';
2. 拷贝文本文件到指定的hdfs目录/HiveDirectory，文件格式满足一行一条记录，字段使用逗号分隔  
   hdfs dfs -copyFromLocal ./goods\_file1.txt /HiveDirectory  
   hdfs dfs -copyFromLocal ./goods\_file2.txt /HiveDirectory
3. 查看表的行数、数据，都反映了文本文件的数据

superset访问hive，实验成功：

hive://hadoop@42.194.XXX.XXX:7001/tutorial?auth=NONE

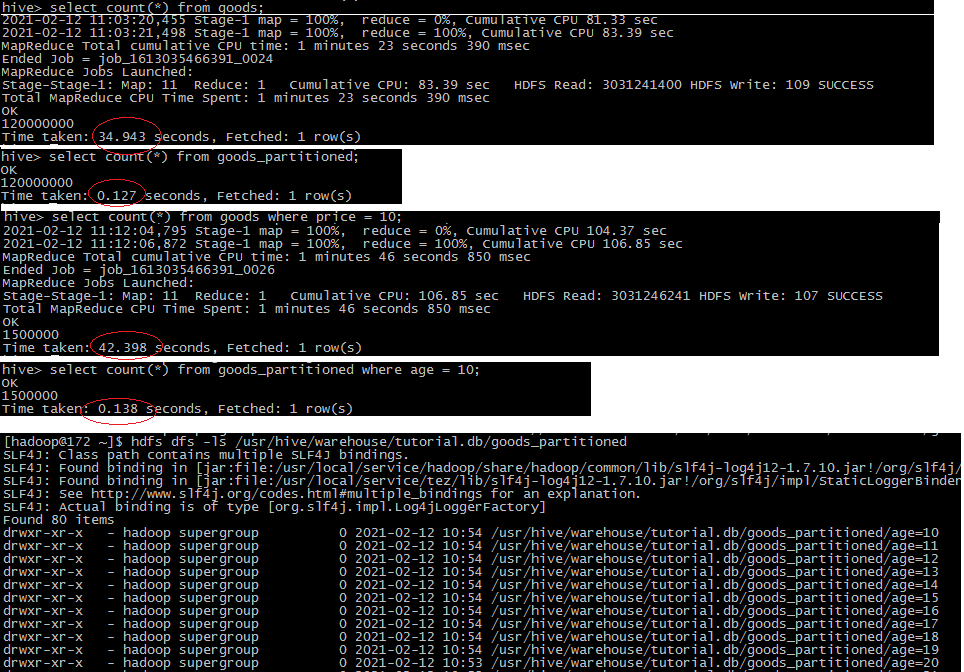
腾讯云EMR集群，详细见：

<https://cloud.tencent.com/document/product/589/42958>

注意确保superset服务器能够联通EMR的master。

HIVE支持partitioned表和clustered表，可以认为手工索引分区，有利于更快的检索。

同构的两张表，一张按照age（写错了，其实是price）字段partitioned，检索快很多：

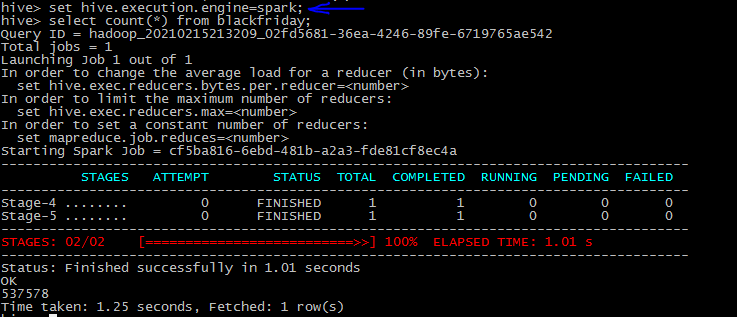


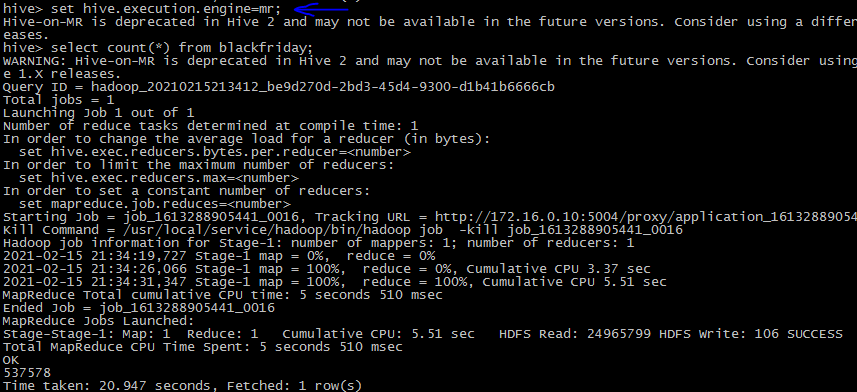
但是我用clustered表，怎么没有明显加速呢？

这个文档不错：

<https://www.javatpoint.com/dynamic-partitioning-in-hive>

另外hive也可以基于spark、tez进行运算的，而且准备废弃对基于MR的支持：

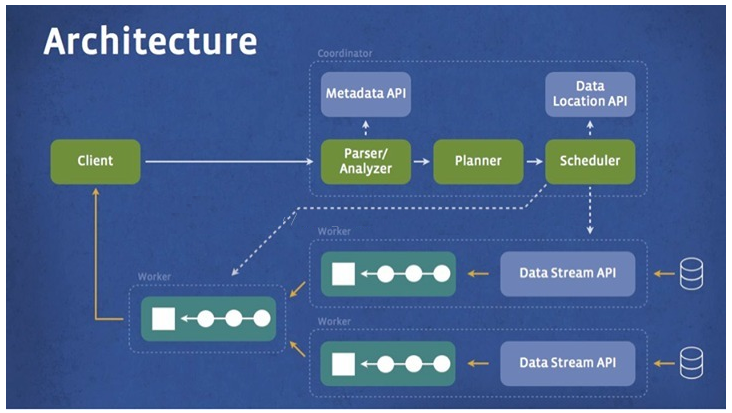




### presto

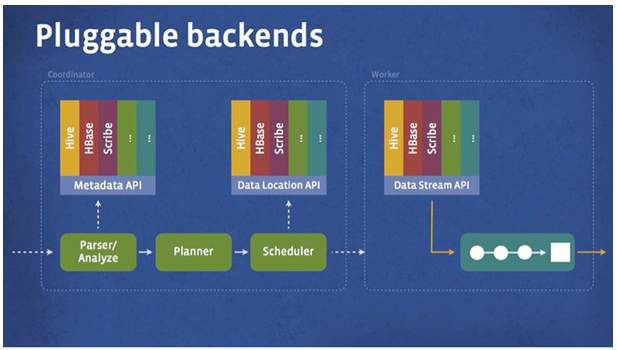
Presto查询引擎是一个Master-Slave的架构，由下面三部分组成:

1. 一个Coordinator节点，负责解析SQL语句，生成执行计划，分发执行任务给Worker节点执行
2. 一个Discovery Server节点，通常内嵌于Coordinator节点中
3. 多个Worker节点，负责实际执行查询任务,负责与HDFS等数据层交互读取数据



Presto设计了一个简单的数据存储的抽象层， 来满足在不同数据存储系统之上都可以使用SQL进行查询

存储插件（连接器,connector）只需要提供实现以下操作的接口， 包括对元数据（metadata）的提取，获得数据存储的位置，获取数据本身的操作等，除了主要使用的Hive/HDFS后台系统之外，也开发了一些连接其他系统的Presto 连接器，包括HBase，Scribe和定制开发的系统。

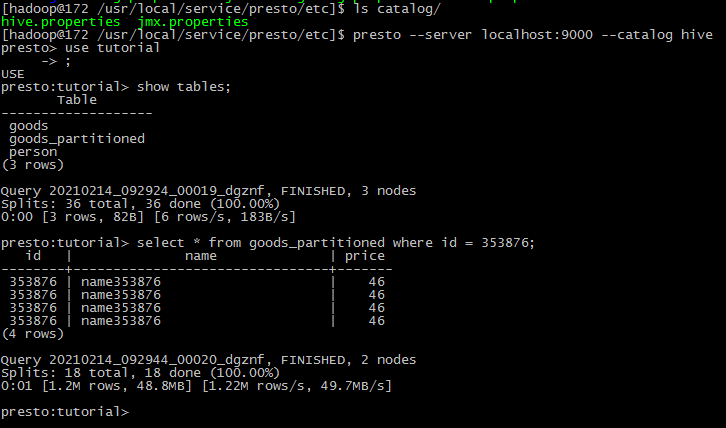


使用ahanaio/prestodb-sandbox docker镜像可以快速搭建一个体验环境，但我用这个presto容器访问昨天的clickhouse什么的都失败了：

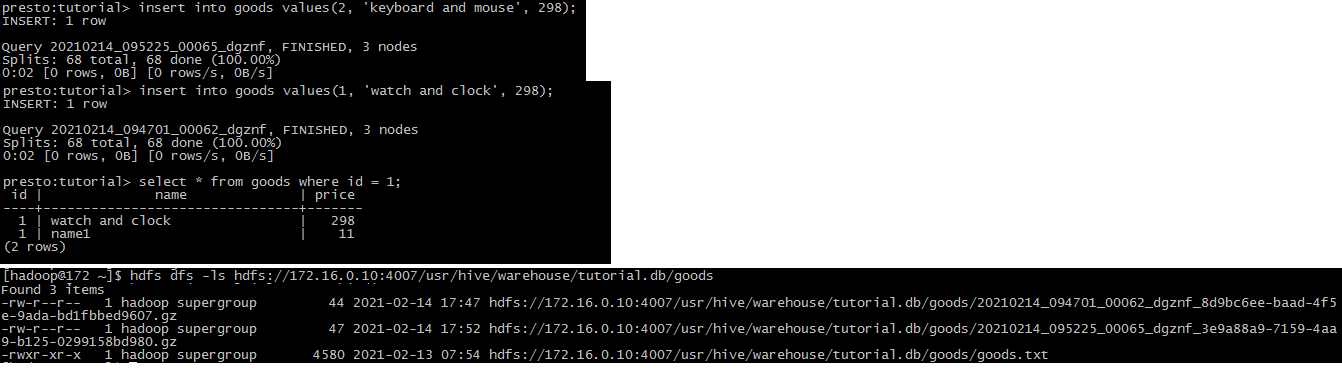
docker run -d -p 8080:8080 -v /presto/presto:/var/lib/presto:rw -v /presto/etc:/opt/presto-server/etc:rw -v /etc/localtime:/etc/localtime:ro --name presto ahanaio/prestodb-sandbox

改为使用腾讯云EMR的presto，很容易上手，详细可见：

<https://cloud.tencent.com/document/product/589/12327>



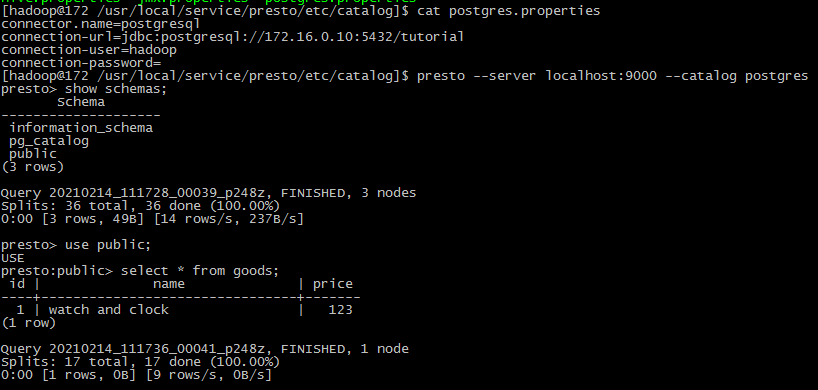
往hive表里每插入一次操作，会产生一个hdfs文件：



我不知死活的尝试配置presto连接一个外部的数据源（postgres为例），还是遇到不少坑的：

1. catalog配置文件的connector.name的值要写对， postgresql，写错了启动不了，提示找不到这样的驱动还是什么
2. postgres要修改postgresql.conf和pg\_hba.conf，监听非127.0.0.1上的端口，并允许来自其他主机的访问
3. presto的每个worker和coordinate节点都要配置catalog。这个很low哈

也算历经千辛万苦，截个图记录一下最后的成功：



### sqoop

sqoop就是一个在sql数据库和hadoop间倒腾数据的工具，因为要应对各种数据源和各种情况，所以参数贼JB复杂，很容易就让程序员觉得老子不用这个傻鸟，这是残疾人用的，老子自己写一个。尤其是hive和sql数据源的用户要求是同一个，因为只有一个参数用来指定用户名：--username。这个太操蛋了。

下面这条语句是从hive的tutorial.goods表导数据到postgres的tutorial.goods表的命令：

sqoop export --connect jdbc:postgresql://localhost/tutorial --table goods --username hadoop --hcatalog-database tutorial --hcatalog-table goods

执行起来很占用资源，键盘都卡死的感觉，过一会就抛出异常了。表里没几条数据都搞不定！

换一条语句，也抛出异常：

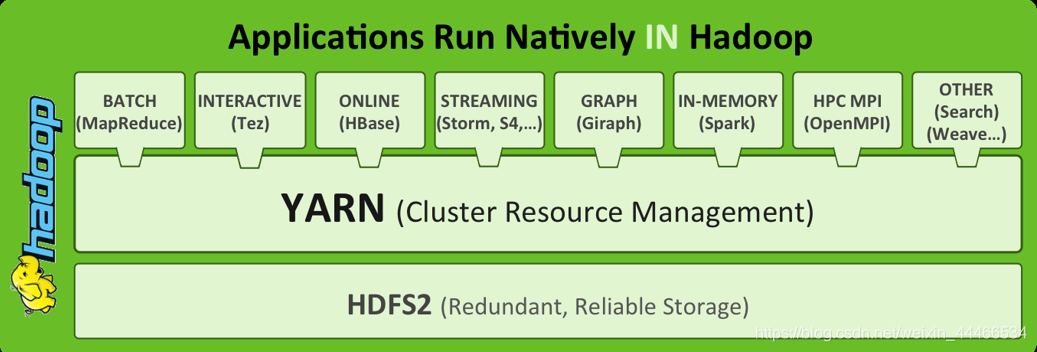
sqoop export --connect jdbc:postgresql://localhost/tutorial --table goods --username hadoop --export-dir /usr/hive/warehouse/tutorial.db/goods

体验失败了

## 2021/2/13 年初二

### YARN

Yarn是通用的资源管理和任务调度器，为上层应用提供统一的资源管理和调度， spark、tez、storm都可基于yarn运行，可以认为yarn是一个分布式操作系统：



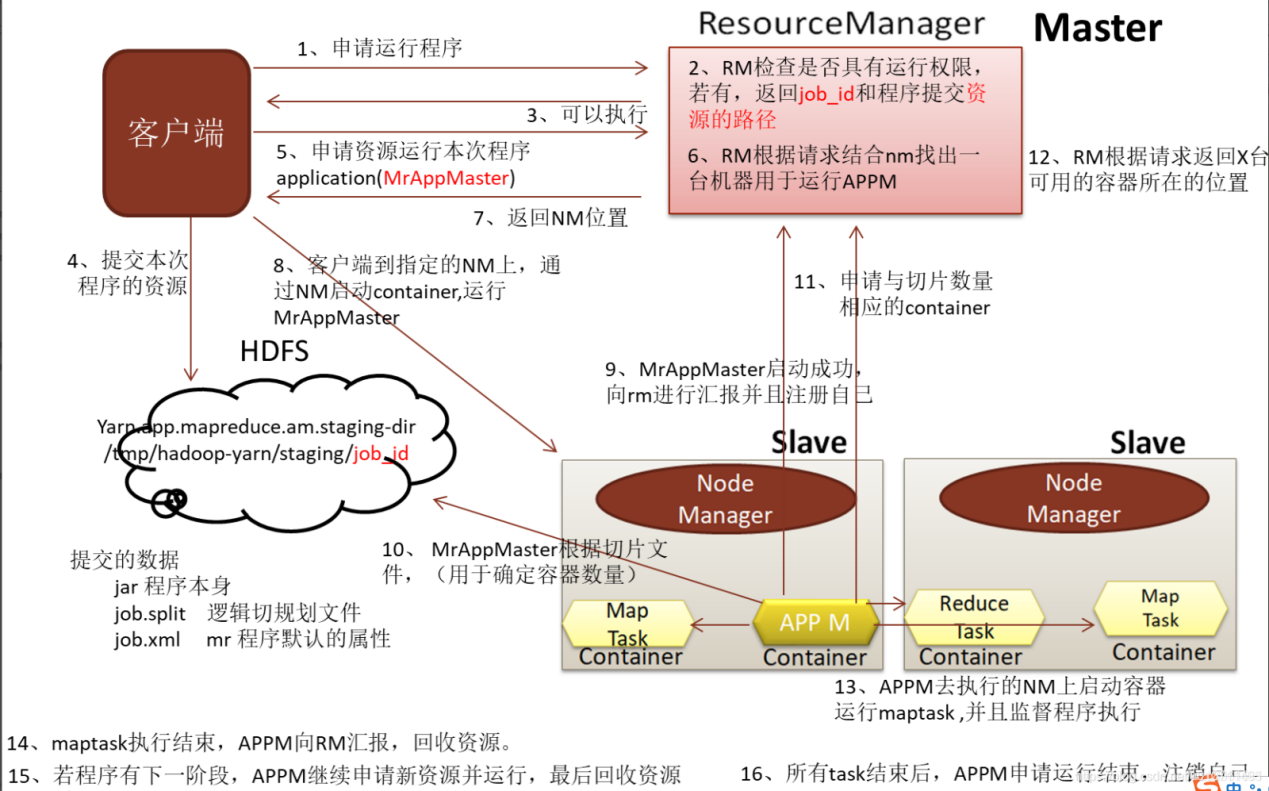
与它类似功能，还有一个叫apache mesos，据说很少用了。

看到这里，我就会想，yarn能否用来调度我们自己的任务呢，例如混流任务。网上一搜，还真有网友做了个hello world demo：

<https://blog.csdn.net/qq_22222499/article/details/105875759>

但代码看着真tmd 闹心，这应该也是yarn在普通群众中未能普及的原因吧。

有热心网友通过下图来展示yarn的工作原理，算是比较清楚的：

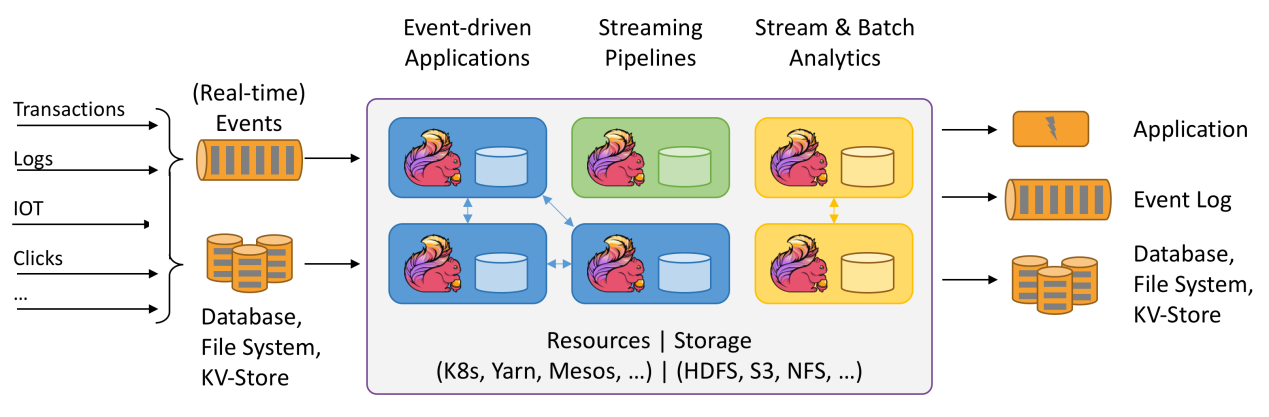


既然Java代码这么闹心，那yarn有没有简洁的python接口呢？

1. 这个算是star比较多的，但也只提供了查询运行信息的接口：  
   <https://github.com/CODAIT/hadoop-yarn-api-python-client>
2. Skein，看起来不错，可以调度和查询，但没有能够安装成功：  
   <https://github.com/jcrist/skein>  
   <https://jcristharif.com/skein/quickstart.html>

### Flink

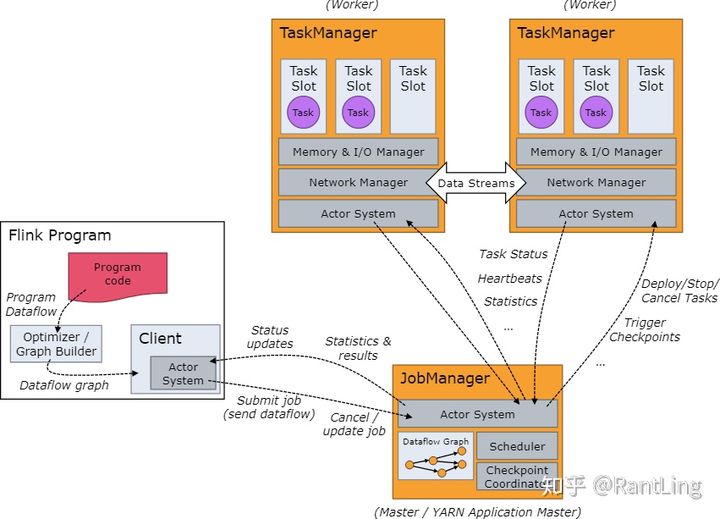
Flink是类似spark这样的计算框架，似乎更侧重流式计算。中间的小松鼠就是flink：



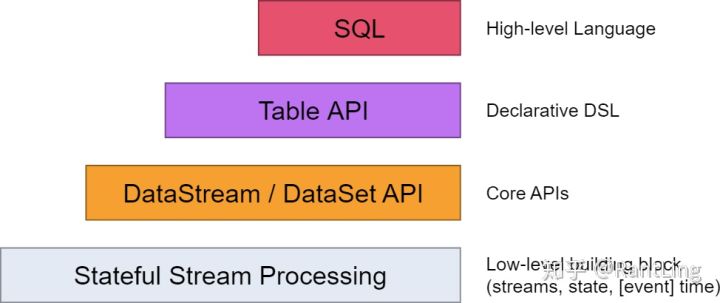
Spark、storm、flink三者在微信指数上对比如下，可见spark还是老大，而Tez词条干脆没有被收录：



根据渊博的知乎网友，flink的架构如下：

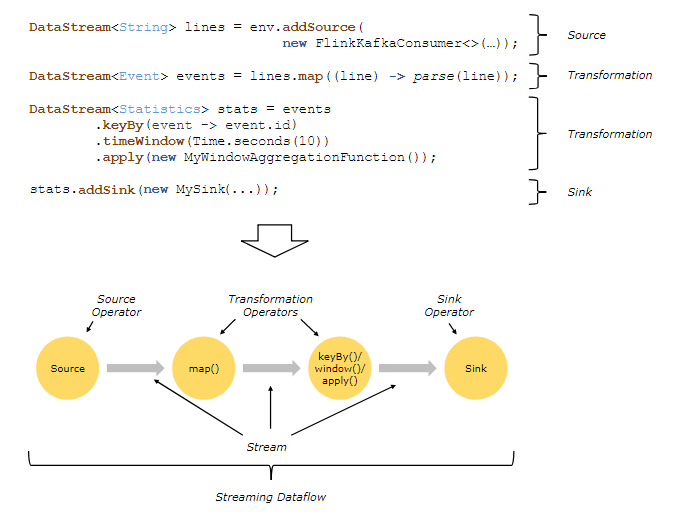


Spark的编程接口主要是围绕RDD和DataFrame展开的，而Flink提供了四层不同抽象程度的api接口，越往上越人性化，越往下越灵活：



第三层所示：DataStream就是流式计算，DataSet就是批处理。

下面这张图是flink数据处理的一个例子：

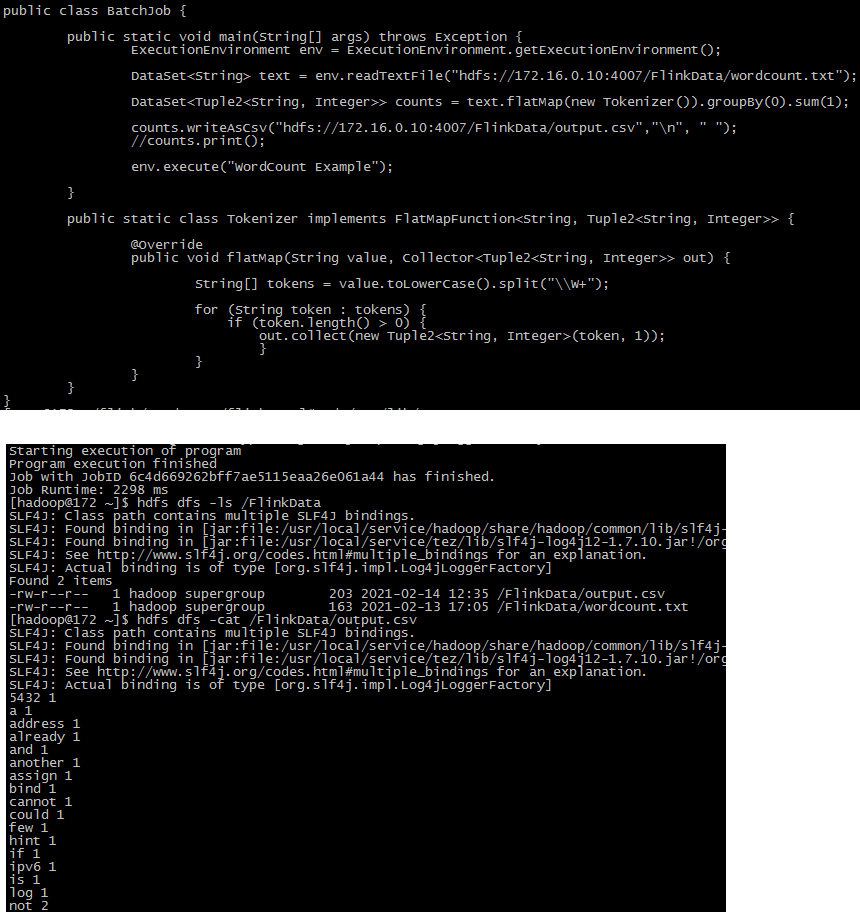


按照网友的分享，可以快速体验一下wordcount例子，感觉flink编程和spark很类似：

https://ci.apache.org/projects/flink/flink-docs-release-1.12/zh/dev/batch/examples.html <https://github.com/wangzhiwubigdata/God-Of-BigData>

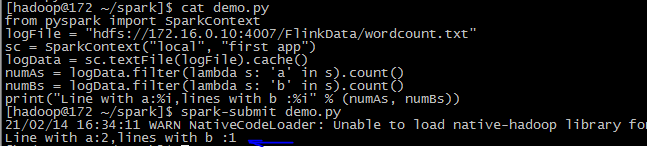
会遇到哪些坑呢：

1. flink run执行的时候，注意用-c指定main class：  
   flink run -c com.bd.flink.BatchJob target/flink-pro-1.0.jar
2. 报错 could not retrieve the execution result:  
   flink不识别hadoop环境，因此再将flink与hadoop整合时，需要将一个jar包放入flink的lib目录下  
   具体的jar包为 flink-shaded-hadoop-2-uber-2.7.5-10.0.jar 具体的jar包版本需要根据安装的hadoop来选择，2.7，2.6等  
   jar包下载连接 <https://flink.apache.org/downloads.html>
3. 执行flink的用户要与hdfs、hadoop的用户一致，我是用hadoop用户。



### Spark

顺便勾选安装了一下spark组件，用pyspark的一个demo简单验证了：



<https://www.jianshu.com/p/5a42fe0eed4d>

之前用过基于RDD的java接口做过排序、洗牌啥的小实验，就不重复体验了。

## 2021/2/14 年初三

### Kylin

Tmd官网说半天也没有说清楚kylin干啥的，无所不知的知乎网友一针见血：

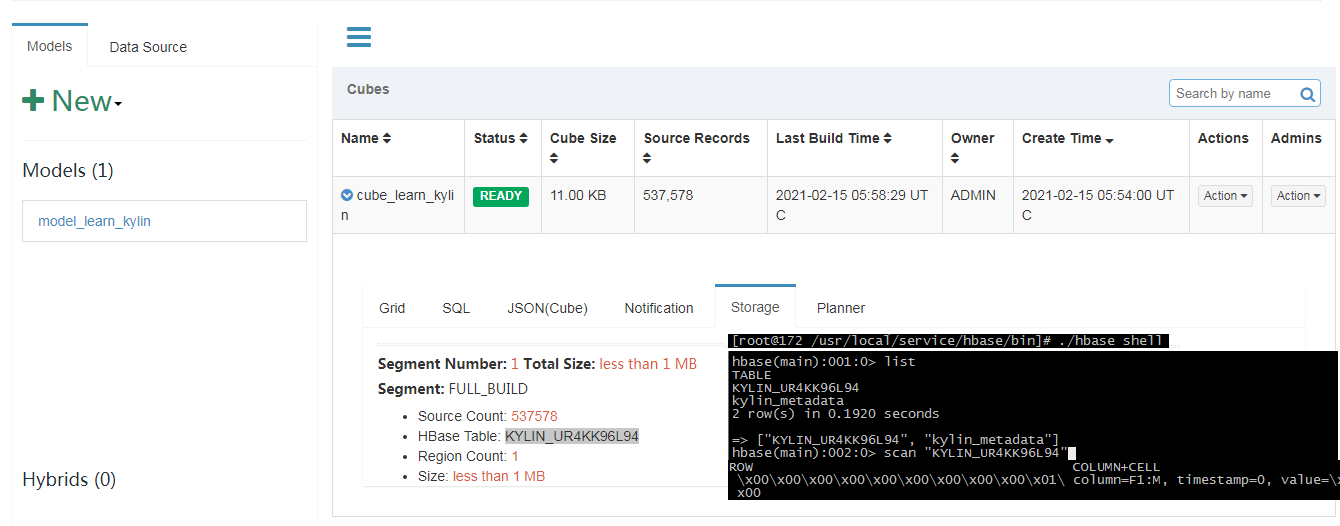
不同于“大规模并行处理”（Massive Parallel Processing，MPP）架构的Hive、Presto等，Apache Kylin采用“预计算”的模式，用户只需要提前定义好查询维度，Kylin将帮助我们进行计算，并将结果存储到HBase中，为海量数据的查询和分析提供亚秒级返回，是一种典型的“空间换时间”的解决方案。

将数据源Hive表中的数据按照指定的维度和指标 由计算引擎MapReduce离线计算出所有可能的查询结果(即Cube)存储到HBase中。

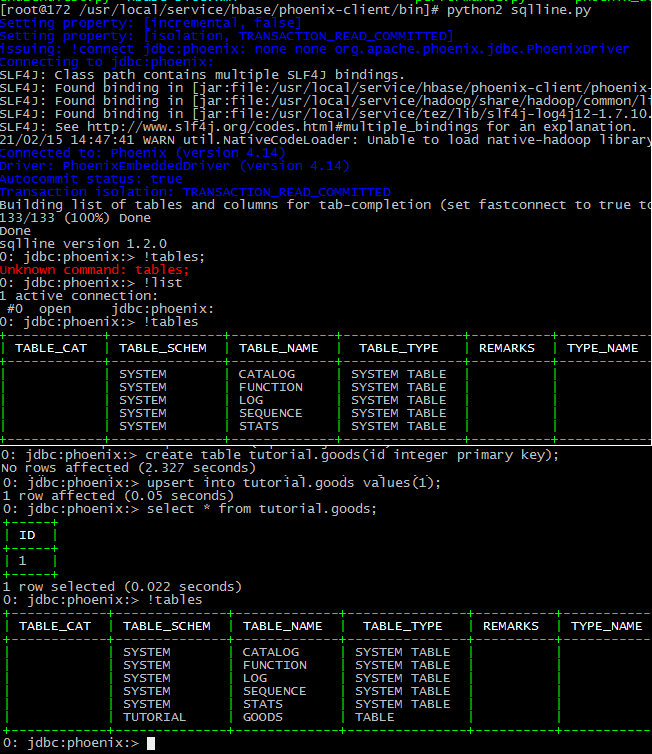
值得一提的是，Kylin对数据源、执行引擎和Cube存储三个核心模块提取出了抽象层，这意味着这三个模块可以被任意地扩展和替换。比如可以使用Spark替代MapReduce作为Cube的构建引擎，使用Cassandra替代HBase作为Cube计算后数据的存储等。

跟着官网的文档比较容易的走通流程：

<https://kylin.apache.org/docs/tutorial/web.html>



表里的内容好像不对，用phoenix去访问也访问不到：

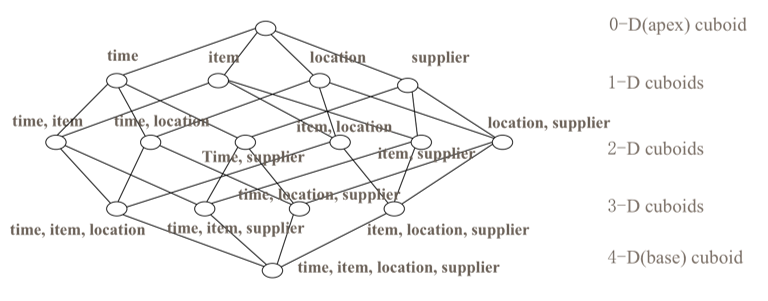


### Cube

数据立方不是一个组件，是一种加速查询的数据处理方式，典型的空间换时间的思路：

借用网友的一段文字：

cube是所有dimession的组合，每一种dimession的组合称之为cuboid。某一有n个dimession的cube会有2n个cuboid，如图：



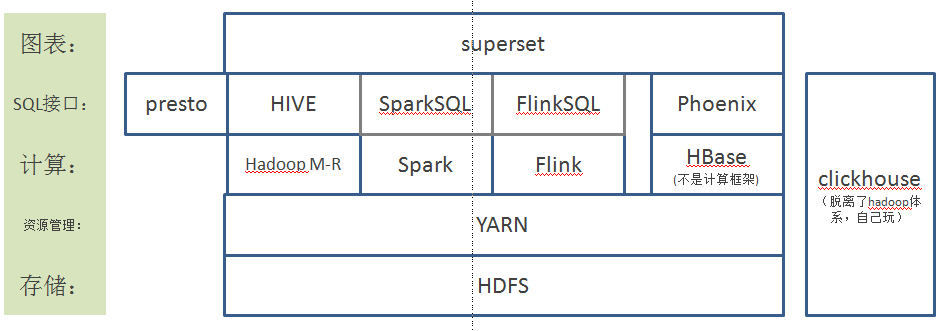
对应一张hive表，有time,item,location,supplier这四个维度，则0-D cuboid时对应的查询语句为 select sum(money) from table;1-D cuboid对应的查询语句有四个，分别为select sum(money) from table group by time，以及select sum(money) from table group by item,以及select sum(money) from table group by location。对应的在2-D时group by 后面的维度会是time,item,location,supplier两两组合。如果不采取优化措施，理论上kylin在预计算过程中会对上述每一种组合进行预计算，随着维度的增加，计算量将会呈几何倍数的增长。为了解决这种问题，kylin对dimession做了分类

<https://www.cnblogs.com/bigdataer/p/6707937.html>

## 2021/2/15 年初四

### 相互关系与定位

以一个图来小结一下前面涉及到组件的位置和相互关系。



基本的原理（不求甚解的话）很简单：  
  
1、分布式文件HDFS负责底层存储，  YARN负责资源调度  
2、在上面是通用计算引擎：有spark，它是用RDD这种数据集合的逻辑抽象，支持filter、交集等操作，比较底层的接口； 或者hadoop MR，提供Map / reduce原子操作，都比较底层 ； 对等的还有个tez，flink  
3、HIVE基于spark/hadoop mr/tez，提供更友好的SQL界面； presto phoenix impala也是解决这个问题  
  
4、superset grafana tableu就是搞定图形化展示

5、clickhouse/HBase+phoenix两者比较等价，是SQL方式的存储为主的东西，也有一些计算的能力，相比spark/flink这些，计算的通用性不够，局限在SQL形式的计算

腾讯云EMR还提供了很多组件，假期要过完了，来不及一一体验：



如果不是直接使用公有云的EMR套装，自己裸机安装的话，不知道要折腾到怎么个吐血情况。