Apache Kylin

# 1.概览

Apache Kylin™是一个开源的分布式分析引擎，提供Hadoop之上的SQL查询接口及多维分析（OLAP）能力以支持超大规模数据，最初由eBay Inc. 开发并贡献至开源社区。它能在亚秒内查询巨大的Hive表， 其能够实现以下各项特性：

1）规模化环境下的极速OLAP引擎：其设计目的在于削减Hadoop环境中处理超过百亿行数据时的查询延迟时间

2）Hadoop上的ANSI SQL接口：Kylin能够在Hadoop之上提供ANSI SQL并支持大部分ANSI SQL查询

3）交互式查询功能：用户可以通过Kylin以秒级以下延迟水平实现与Hadoop数据的交互，在面对同一套数据集使，其性能表现优于Hive查询机制

4）利用MOLAP cube（立方体）对数百亿行数据进行查询，用户能够在Kylin当中定义一套数据模型对其进行预构建，其中所能包含的原始数据记录可超过百亿行

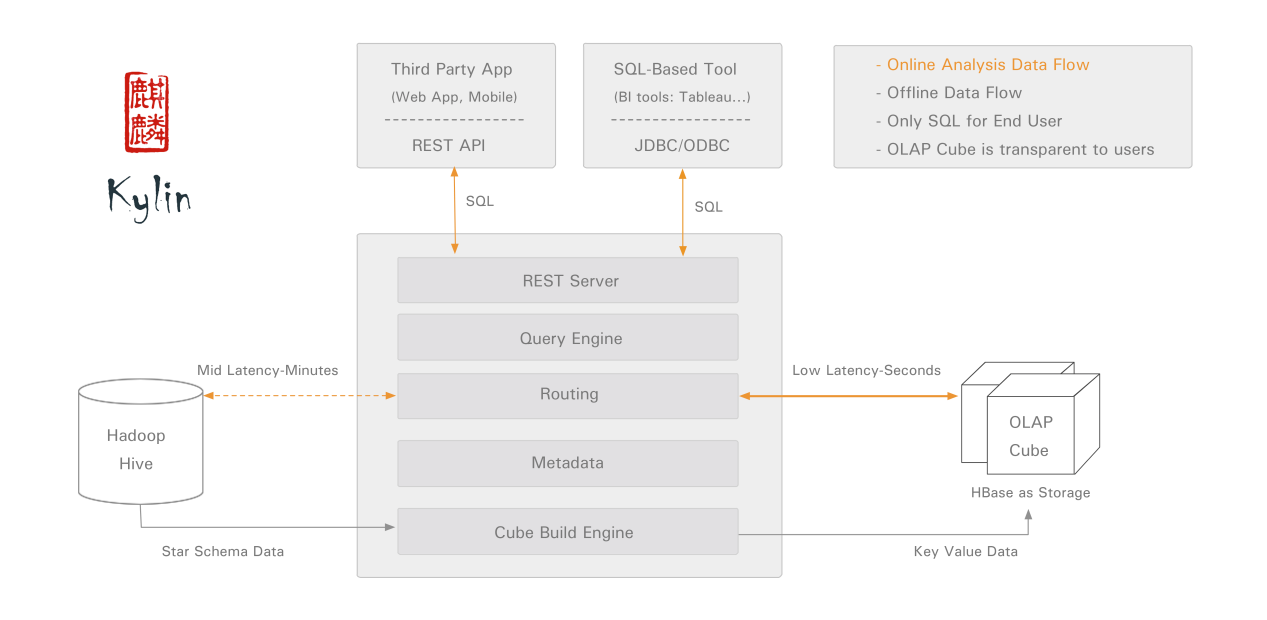
5）与商务智能工具进行无缝化集成：Kylin目前能够与多种智能工具相集成，包括Tableau以及其他第三方应用程序

6）开源ODBC驱动程序，Kylin的ODBC驱动程序从零开始逐步构建而成，而且能够与Tableau实现良好的协作效果。

其他特性包括：任务管理和监控机制，通过压缩与编码机制降低存储容量需求，Cube的增量式更新，利用HBase协处理器实现查询延迟控制，对不同计数进行近似查询的能力，提供易于使用的Web界面（旨在对cube进行管理、构建、监控与查询），cube/项目层面对ACL进行设置的安全功能，支持LDAP集成。

# 2.架构

下图中勾勒出Cube构建引擎（Cube Build Engine）是如何以离线处理方式将关系型数据库转化成键值数据的，其中黄线部分表现出在线分析数据的处理流程。数据请求可以利用基于 SQL的工具由SQL提交产生，或者利用第三方应用程序通过Kylin的RESTful服务来实现。RESTful服务会调用Query Engine，后者则检测对应的目标数据集是否真实存在。如果确实存在，该引擎会支持访问目标数据并以次秒级延迟返回结果。如果目标数据集并不存在，该引擎则会根据设计将无匹配数据集的查询路由至Hadoop上的SQL处，即交由Hive等Hadoop集群负责处理。



Kylin平台内的组件如下：

1）元数据管理工具（Metadata Manager），Kylin是一款元数据驱动型应用程序，元数据管理工具是一大关键组件，用户对保存在Kylin当中的元数据进行管理，其中包括最为重要的cube元数据。其他全部组件的正常运行都需以元数据管理工具为基础。

2）任务引擎（Job Engine），这套引擎的涉及目的在于处理所有离线任务，其中包括Shell脚本、JAVA API以及Map Reduce任务等等。任务引起对Kylin当前中的全部任务加以管理及协调，从而确保每一项任务都能得到切实执行并解决其间出现的故障

3）存储引擎（Storage Engine），这套引擎负责管理底层存储，特别是cuboid，其以键值对的形式进行保存。存储引擎适用的是HBase，Kylin能够通过扩展来实现对其它键值系统的支持，例如Redis

4）REST Server，面向应用程序开发的入口点，旨在实现针对Kylin平台的应用开发工作，此类应用程序可以提供查询、获取结果，触发cube构建任务，获取元数据以及获取用户权限等

5）ODBC驱动，支持第三方工具与应用程序，例如Tableau，构建了一套ODBC驱动程序并对其进行开源。

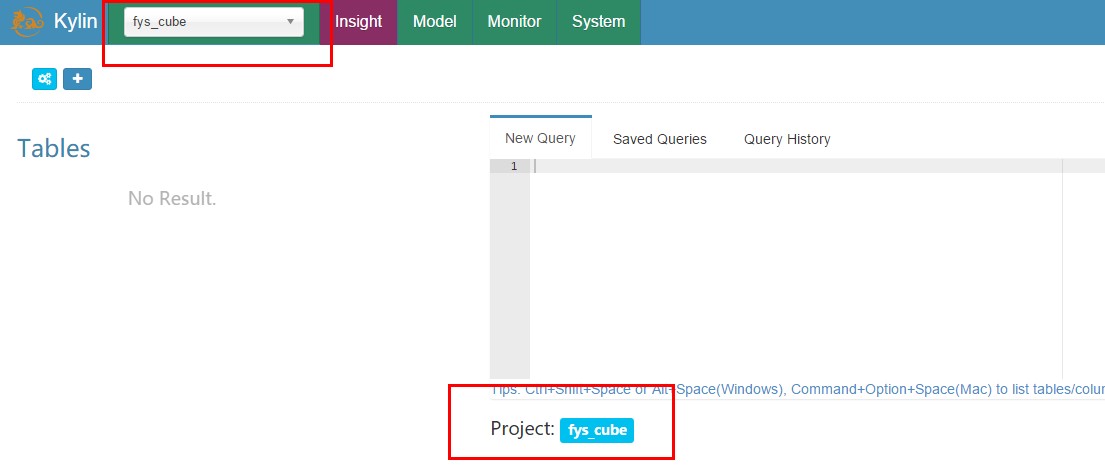
6）查询引擎，当Cube准备就绪后，查询引擎能够获取并解析用户查询，随后与系统中其他组件进行交互，从而向用户返回对应的结果

# 3.Kylin的使用

1）初次登陆kylin，需要新建一个project，才能利用Kylin为我们做OLAP分析，如下所示，点击左上角的加号（Add Project）

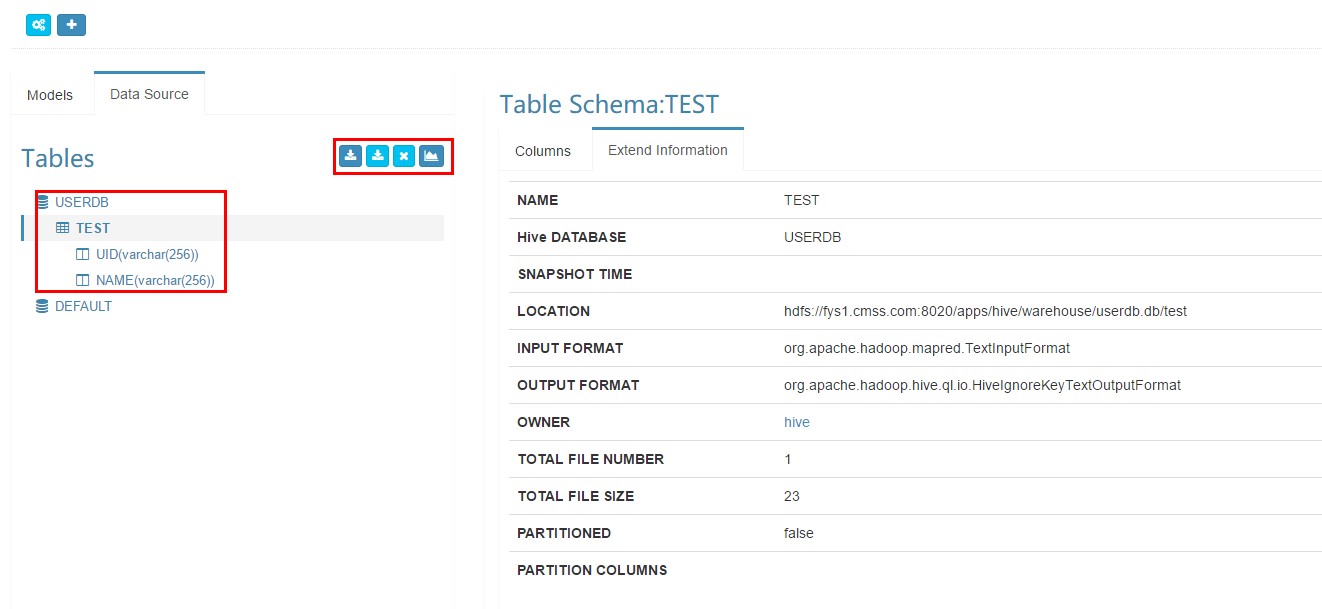


输入project相应信息，后创建project，如下所示：



2）Datasource

存在与Hive中的数据库会同步显示到kylin web页面的Datasource下，选中某个数据库下的某张表，在页面右侧即会显示该表的字段信息，下图以userdb，表test为例，右上的四个框是加载表的操作：



四个表操作，分别是Load Hive Table，Load Hive Table From Tree,Add Streaming Table及Add Streaming Table。

3）Sample Cube

Kylin提供了一个脚本来创建sample cube，该脚本会创建三个hive sample表，执行如下所示：

1. 执行脚本

*${KYLIN\_HOME}/bin/sample.sh*

执行结果如下所示：

*KYLIN\_HOME is set to ./apache-kylin-1.6.0-hbase1.x-bin/bin/../*

*Going to create sample tables in hive...*

*Loading sample data into HDFS tmp path: /tmp/kylin/sample\_cube/data*

*scan complete in 3ms*

*Connecting to jdbc:hive2://fys1.cmss.com:2181,fys2.cmss.com:2181,fys3.cmss.com:2181/default;*

*serviceDiscoveryMode=zookeeper;zooKeeperNamespace=hiveserver2*

*Connected to: Apache Hive (version 1.1.0-bc1.3.4)*

*Driver: Hive JDBC (version 1.1.0-bc1.3.4)*

*Transaction isolation: TRANSACTION\_REPEATABLE\_READ*

*0: jdbc:hive2://fys1.cmss.com:2181,fys2.cmss.> --*

*…………………………………………………………………………………………………*

*[main] client.HBaseAdmin: Created kylin\_metadata*

*Closing master protocol: MasterService*

*Sample cube is created successfully in project 'learn\_kylin'.*

*Restart Kylin server or reload the metadata from web UI to see the change.*

在hive中创建的表如下所示：

*0: jdbc:hive2://fys1.cmss.com:2181,fys2.cmss.> show tables;*

*+---------------------------+--+*

*| tab\_name |*

*+---------------------------+--+*

*| kylin\_cal\_dt |*

*| kylin\_category\_groupings |*

*| kylin\_country |*

*| kylin\_sales |*

*+---------------------------+--+*

在hbase中创建的表如下：

*hbase(main):001:0> list*

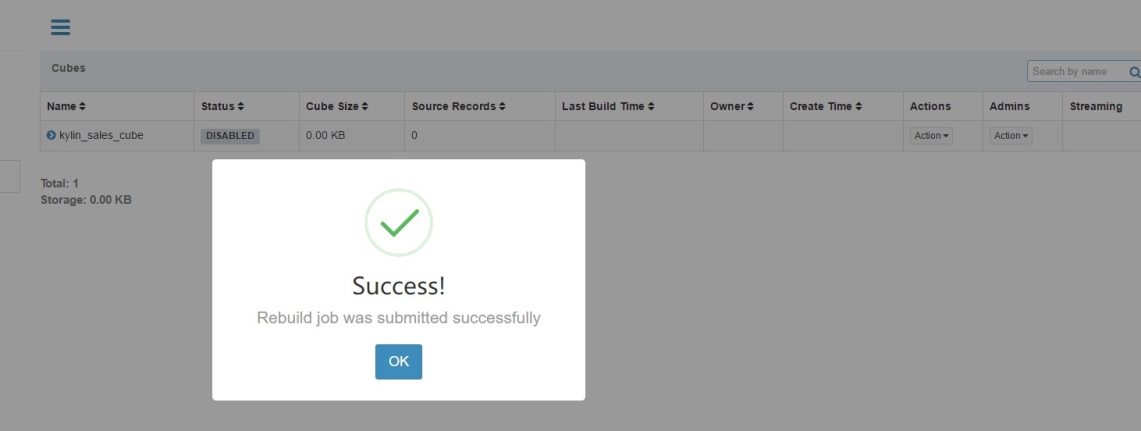
*TABLE*

*kylin\_metadata*

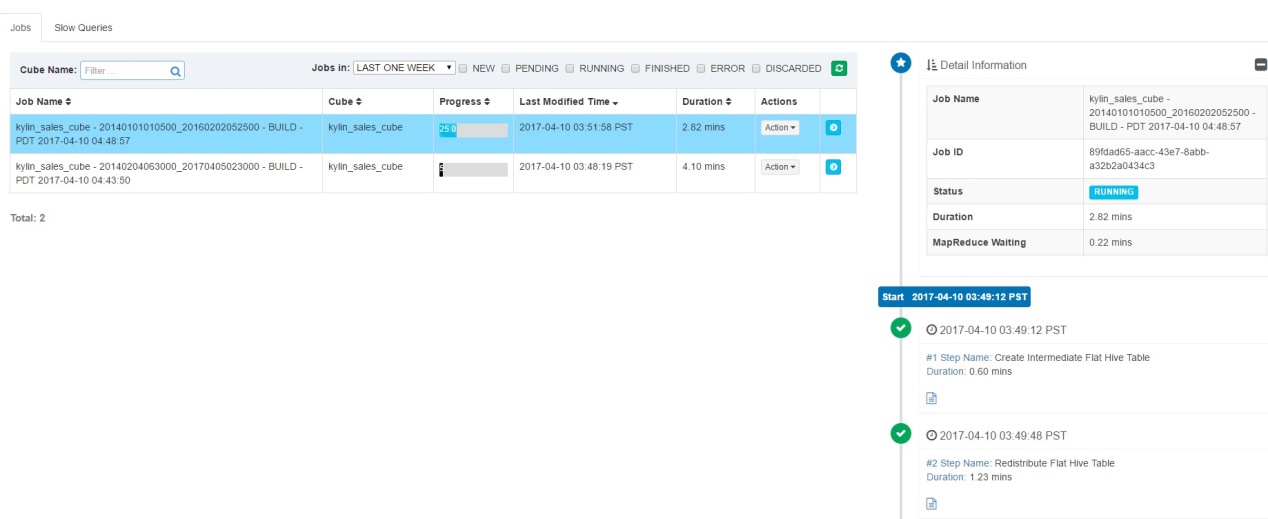
*kylin\_metadata\_acl*

*kylin\_metadata\_user*

1. 重新启动kylin server后，可以在kylin web上查找到project learn\_kylin
2. 选择sample cube ，kylin\_sales\_cube，点击ACTIONS => BUILD，选择的日期是晚于2014-1-1



1. 点击monitor，可以看到cube执行进度



执行完成后，hive的中间表会被清空。但是Hbase中增加表：

*hbase(main):001:0> list*

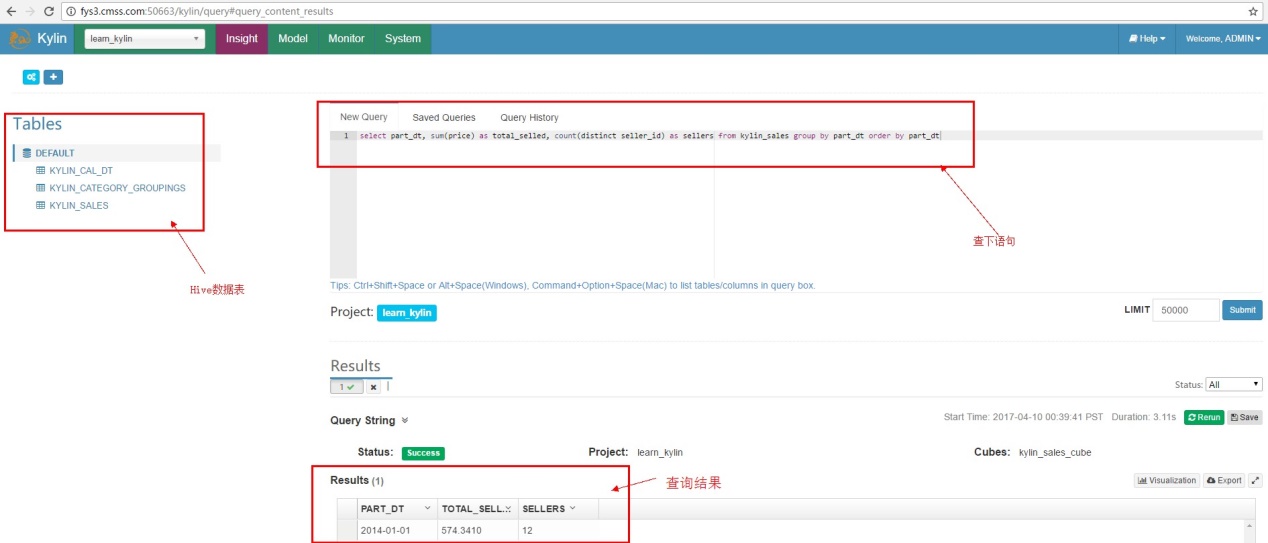
*TABLE*

*KYLIN\_0V7K6VO5S4*

1. 在Insight中执行SQLS，例如：

*select part\_dt, sum(price) as total\_selled, count(distinct seller\_id) as sellers from kylin\_sales group by part\_dt order by part\_dt*

查询结果如下：

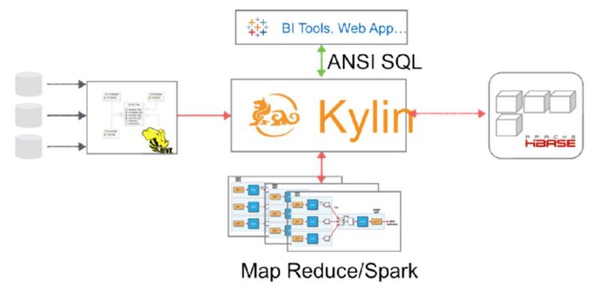


http://kylin.apache.org/docs20/tutorial/kylin\_sample.html

# 4.Kylin执行过程

Kylin的核心思想是与计算，理论基础是以空间换时间，即针对一个星型拓扑结构的数据立方体，预算多个维度组合的度量，然后将结果保存到hbase中，对外暴露JDBC/ODBC/Rest api查询接口，把高复杂度的聚合计算，多表连接等操作转换成对预算结果的查询，实现实时查询。

数据立方体，一般由Hive中的一个事实表，多个查找表组成，预计算的过程即Cube的build过程，如下图所示：



1. Kylin从用来做数据仓库的Hive中读取源数据
2. 使用MapReduce作为cube构建引擎
3. 把计算结果保存到HBase中，并对外暴露Restful API/JDBC/ODBC等查询接口

# 附录.各模块中的相关数据

1）HBASE中创建的表

hbase(main):001:0> list

TABLE

KYLIN\_NK9GP80GS1

kylin\_metadata

kylin\_metadata\_acl

kylin\_metadata\_user

配置项：

# The metadata store in hbase

kylin.metadata.url=kylin\_metadata@hbase

2）Hive中创建的表

jdbc:hive2://10.133.46.155:10000/default> show tables;

+-------------------------------------------------------------------------+--+

| tab\_name |

+-------------------------------------------------------------------------+--+

| kylin\_cal\_dt |

| kylin\_category\_groupings |

| kylin\_intermediate\_kylin\_sales\_cube\_desc\_20120101000000\_20161102010500 |

| kylin\_intermediate\_kylin\_sales\_cube\_desc\_20161102051000\_20161110021000 |

| kylin\_intermediate\_kylin\_sales\_cube\_desc\_20161102051000\_20161110073500 |

| kylin\_sales |

# Hive database name for putting the intermediate flat tables

kylin.job.hive.database.for.intermediatetable=default

3）Hdfs中的数据

# Temp folder in hdfs, make sure user has the right access to the hdfs directory

kylin.hdfs.working.dir=/kylin

[hdfs@sgserver006 ~]$ hdfs dfs -ls /kylin/kylin\_metadata

Found 9 items

/kylin/kylin\_metadata/coprocessor

/kylin/kylin\_metadata/kylin-5738fe63-b58e-4d97-9791-702bc4112207

/kylin/kylin\_metadata/kylin-5f226f19-7975-4ff7-9415-aaf0d88135c4

/kylin/kylin\_metadata/kylin-89975fe2-996f-47f8-aafa-54e58822f0ae

/kylin/kylin\_metadata/kylin-89b14410-2b2b-4d25-812f-dc72b8ec72ec

/kylin/kylin\_metadata/kylin-8b828ecf-850f-42b6-8b69-358082c75d82

/kylin/kylin\_metadata/kylin-ccd9f050-6d3e-47a3-ae95-0f09d614def4

/kylin/kylin\_metadata/kylin-d65c2c7f-9e42-4ee1-9013-d56843c527df

/kylin/kylin\_metadata/kylin-dd7fda29-f928-4f8a-8ca5-72399066666e

4）MR执行的临时目录

/user/kylin/.staging

参考文献：

http://www.csdn.net/article/2014-10-25/2822286