Impala

# Impala的基本概念

## 1.1 什么是Impala

Cloudera公司推出，提供对HDFS、Hbase数据的高性能、低延迟的交互式SQL查询功能。

基于Hive，使用内存计算，兼顾数据仓库、具有实时、批处理、多并发等优点。

是CDH平台首选的PB级大数据实时查询分析引擎。



## 1.2 Impala的优缺点



## 1.3 Impala的组成



## 1.4 Impala的运行原理



Impala执行查询的具体过程：

1）当用户提交查询前，Impala先创建一个负责协调客户端提交的查询的Impalad进程，该进程会向Impala State Store提交注册订阅信息，State Store会创建一个statestored进程，statestored进程通过创建多个线程来处理Impalad的注册订阅信息。

2）用户通过CLI客户端提交一个查询到impalad进程，Impalad的Query Planner对SQL语句进行解析，生成解析树；然后，Planner把这个查询的解析树变成若干PlanFragment，发送到Query Coordinator.

3）Coordinator通过从元数据库中获取元数据，从HDFS的名称节点中获取数据地址，以得到存储这个查询相关数据的所有数据节点。

4）Coordinator初始化相应impalad上的任务执行，即把查询任务分配给所有存储这个查询相关数据的数据节点。

5）Query Executor通过流式交换中间输出，并由Query Coordinator汇聚来自各个impalad的结果。

1. Coordinator把汇总后的结果返回给CLI客户端。
2. Impala的安装

## 2.1 Impala的地址

1. Impala的官网

<http://impala.apache.org/>

1. Impala文档查看

<http://impala.apache.org/impala-docs.html>

1. 下载地址

<http://impala.apache.org/downloads.html>

## 2.2 Impala的安装方式

Impala有两种安装方式：

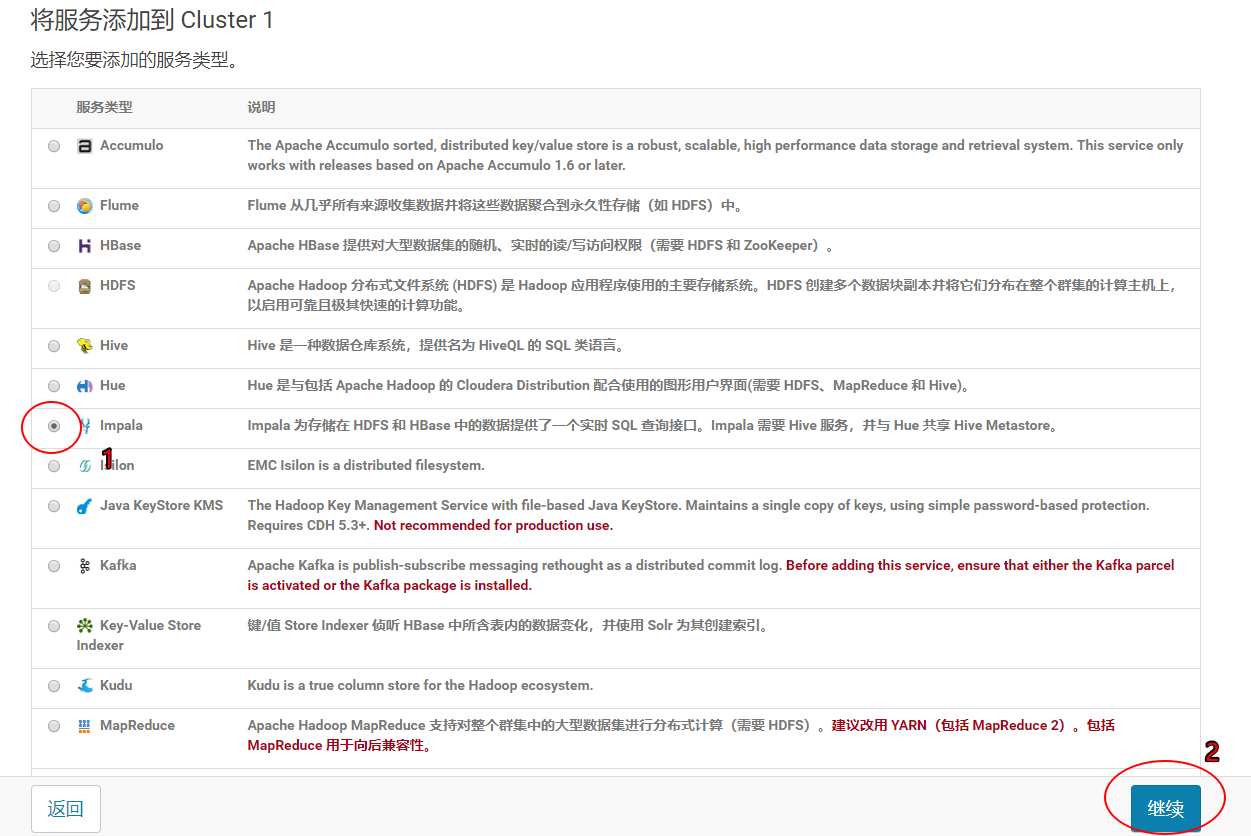
1. 手动安装。
2. CDH安装（推荐）。

下面我们使用Cloudera Manager安装Impala：

1）在主页中点击添加服务



2）选择Impala服务

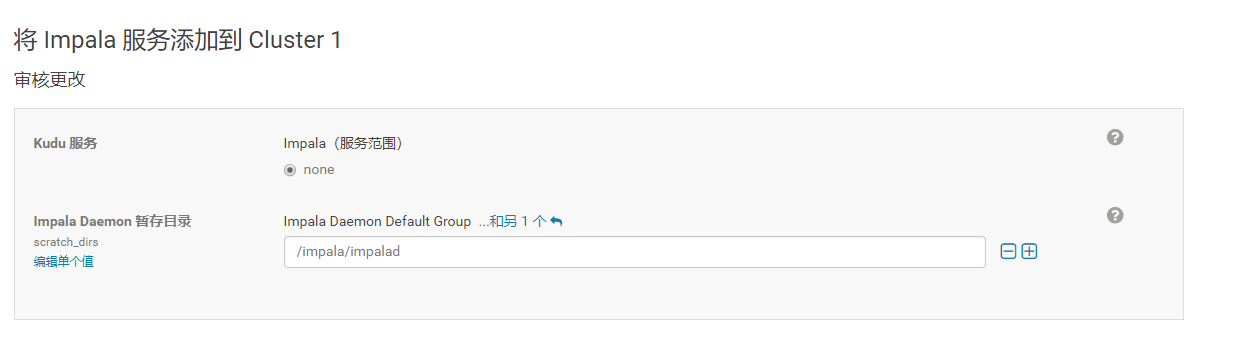


3）进行角色分配

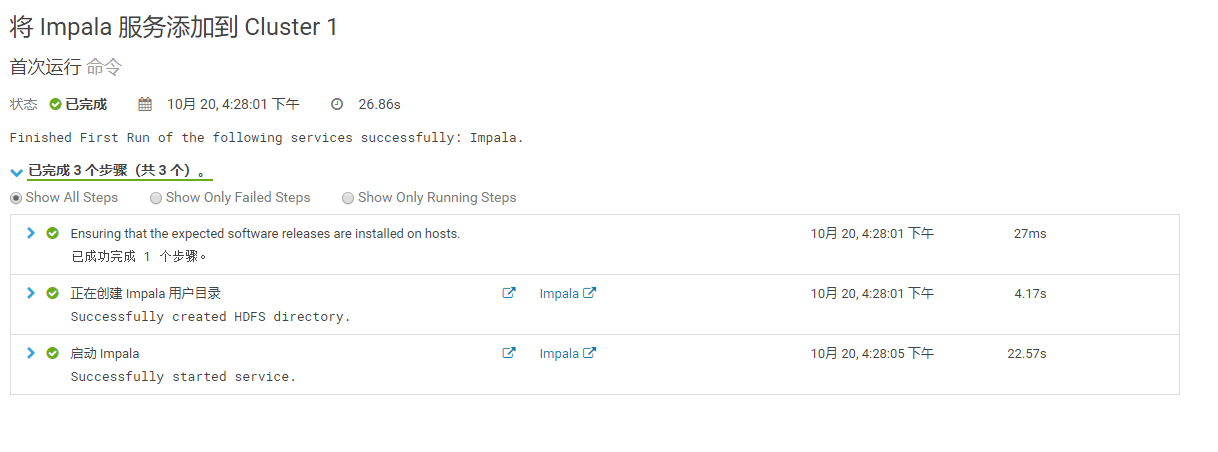


注意：最好将StateStore和CataLog Sever单独部署在同一节点上。

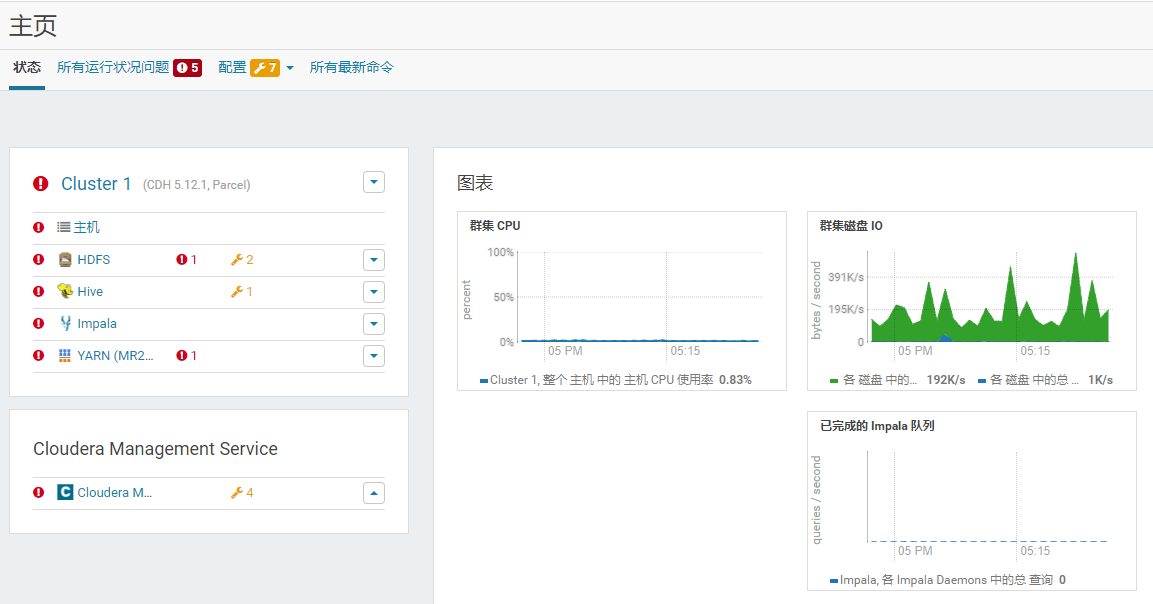
4）配置Impala



5）启动Impala



6）安装成功



2.3 Impala 的监护管理

　　可以通过下面的链接来访问Impala的监护管理页面：

1．查看StateStore  
　　　　<http://hadoop102:25020/>

　2．查看Catalog

<http://hadoop102:25010/>

2.4 Impala的初体验

1.启动Impala

[root@hadoop102 ~]# impala-shell

2.查看数据库

[hadoop102:21000] > show databases;

3.打开默认数据库

[hadoop102:21000] > use default;

4.显示数据库中的表

[hadoop102:21000] > show tables;

5.创建一张student表

[hadoop102:21000] > create table student(id int, name string)

> row format delimited

> fields terminated by '\t';

6.向表中导入数据

[hadoop103:21000] > load data inpath '/student.txt' into table student;

注意：

1. 关闭（修改hdfs的配置dfs.permissions为false）或修改hdfs的权限，否则impala没有写的权限

[hdfs@hadoop103 ~]$ hadoop fs -chmod 777 /

1. Impala不支持将本地文件导入到表中

7.查询

[hadoop103:21000] > select \* from student;

8.退出impala

[hadoop103:21000] > quit;

1. Impala的操作命令

## 3.1 Impala的外部shell

|  |  |
| --- | --- |
| 选项 | 描述 |
| -h, --help | 显示帮助信息 |
| -v or --version | 显示版本信息 |
| -i hostname, --impalad=hostname | 指定连接运行 impalad 守护进程的主机。默认端口是 21000。 |
| -q query, --query=query | 从命令行中传递一个shell 命令。执行完这一语句后 shell 会立即退出。 |
| -f query\_file, --query\_file= query\_file | 传递一个文件中的 SQL 查询。文件内容必须以分号分隔 |
| -o filename or --output\_file filename | 保存所有查询结果到指定的文件。通常用于保存在命令行使用 -q 选项执行单个查询时的查询结果。 |
| -c | 查询执行失败时继续执行 |
| -d default\_db or --database=default\_db | 指定启动后使用的数据库，与建立连接后使用use语句选择数据库作用相同，如果没有指定，那么使用default数据库 |
| -r or --refresh\_after\_connect | 建立连接后刷新 Impala 元数据 |
| -p, --show\_profiles | 对 shell 中执行的每一个查询，显示其查询执行计划 |
| -B（--delimited） | 去格式化输出 |
| --output\_delimiter=character | 指定分隔符 |
| --print\_header | 打印列名 |

1. 连接指定hadoop103的impala主机

[root@hadoop102 datas]# impala-shell -i hadoop103

1. 使用-q查询表中数据，并将数据写入文件中

[hdfs@hadoop103 ~]$ impala-shell -q 'select \* from student' -o output.txt

1. 查询执行失败时继续执行

[hdfs@hadoop103 ~]$ vim impala.sql

select \* from student;

select \* from stu;

select \* from student;

[hdfs@hadoop103 ~]$ impala-shell -f impala.sql;

[hdfs@hadoop103 ~]$ impala-shell -c -f impala.sql;

1. 在hive中创建表后，使用-r刷新元数据

hive> create table stu(id int, name string);

[hadoop103:21000] > show tables;

Query: show tables

+---------+

| name |

+---------+

| student |

+---------+

[hdfs@hadoop103 ~]$ impala-shell -r

[hadoop103:21000] > show tables;

Query: show tables

+---------+

| name |

+---------+

| stu |

| student |

+---------+

1. 显示查询执行计划

[hdfs@hadoop103 ~]$ impala-shell -p

[hadoop103:21000] > select \* from student;

所谓执行计划，即在查询sql执行之前，先对该sql做一个分析，列出需要完成这一项查询的详细方案：

1. 去格式化输出

[root@hadoop103 ~]# impala-shell -q 'select \* from student' -B --output\_delimiter="\t" -o output.txt

[root@hadoop103 ~]# cat output.txt

1001 tignitgn

1002 yuanyuan

1003 haohao

1004 yunyun

## 3.2 Impala的内部shell

|  |  |
| --- | --- |
| 选项 | 描述 |
| help | 显示帮助信息 |
| explain <sql> | 显示执行计划 |
| profile | (查询完成后执行） 查询最近一次查询的底层信息 |
| shell <shell> | 不退出impala-shell执行shell命令 |
| version | 显示版本信息（同于impala-shell -v） |
| connect | 连接impalad主机，默认端口21000（同于impala-shell -i） |
| refresh <tablename> | 增量刷新元数据库 |
| invalidate metadata | 全量刷新元数据库（慎用）（同于 impala-shell -r） |
| history | 历史命令 |

1. 查看执行计划

explain select \* from student;

1. 查询最近一次查询的底层信息

[hadoop103:21000] > select count(\*) from student;

[hadoop103:21000] > profile;

1. 查看hdfs及linux文件系统

[hadoop103:21000] > shell hadoop fs -ls /;

[hadoop103:21000] > shell ls -al ./;

1. 刷新指定表的元数据

hive> load data local inpath '/opt/module/datas/student.txt' into table student;

[hadoop103:21000] > select \* from student;

[hadoop103:21000] > refresh student;

[hadoop103:21000] > select \* from student;

1. 查看历史命令

[hadoop103:21000] > history；

1. Impala的数据类型

|  |  |  |
| --- | --- | --- |
| Hive数据类型 | Impala数据类型 | 长度 |
| TINYINT | TINYINT | 1byte有符号整数 |
| SMALINT | SMALINT | 2byte有符号整数 |
| INT | INT | 4byte有符号整数 |
| BIGINT | BIGINT | 8byte有符号整数 |
| BOOLEAN | BOOLEAN | 布尔类型，true或者false |
| FLOAT | FLOAT | 单精度浮点数 |
| DOUBLE | DOUBLE | 双精度浮点数 |
| STRING | STRING | 字符系列。可以指定字符集。可以使用单引号或者双引号。 |
| TIMESTAMP | TIMESTAMP | 时间类型 |
| BINARY | 不支持 | 字节数组 |

注意：Impala虽然支持array，map，struct复杂数据类型，但是支持并不完全，一般处理方法，将复杂类型转化为基本类型，通过hive创建表。

1. DDL数据定义

## 5.1 创建数据库

1.创建命令

CREATE DATABASE [IF NOT EXISTS] database\_name

[COMMENT database\_comment]

[LOCATION hdfs\_path];

注：Impala不支持WITH DBPROPERTIE…语法

2. 错误演示

[hadoop103:21000] > create database db\_hive

> WITH DBPROPERTIES('name' = 'ttt');

Query: create database db\_hive

WITH DBPROPERTIES('name' = 'ttt')

ERROR: AnalysisException: Syntax error in line 2:

WITH DBPROPERTIES('name' = 'ttt')

^

Encountered: WITH

Expected: COMMENT, LOCATION

## 5.2查询数据库

### 5.2.1显示数据库

[hadoop103:21000] > show databases;

[hadoop103:21000] > show databases like 'hive\*';

Query: show databases like 'hive\*'

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

| name | comment |

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

| hive\_db | |

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

[hadoop103:21000] > desc database hive\_db;

Query: describe database hive\_db

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

| name | location | comment |

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

| hive\_db | | |

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

### 5.2.2删除数据库

[hadoop103:21000] > drop database hive\_db;

[hadoop103:21000] > drop database hive\_db cascade;

注：

Impala不支持alter database语法

当数据库被 USE 语句选中时，无法删除

## 5.3创建表

### 5.3.1 管理表

[hadoop103:21000] > create table if not exists student2(

> id int, name string

> )

> row format delimited fields terminated by '\t'

> stored as textfile

> location '/user/hive/warehouse/student2';

[hadoop103:21000] > desc formatted student2;

### 5.3.2 外部表

[hadoop103:21000] > create external table stu\_external(

> id int,

> name string)

> row format delimited fields terminated by '\t' ;

## 5.4分区表

### 5.4.1 创建分区表

[hadoop103:21000] > create table stu\_par(id int, name string)

> partitioned by (month string)

> row format delimited

> fields terminated by '\t';

### 5.4.2 向表中导入数据

[hadoop103:21000] > alter table stu\_par add partition (month='201810');

[hadoop103:21000] > load data inpath '/student.txt' into table stu\_par partition(month='201810');

[hadoop103:21000] > insert into table stu\_par partition (month = '201811')

> select \* from student;

注意：

如果分区没有，load data导入数据时，不能自动创建分区。

### 5.4.3 查询分区表中的数据

[hadoop103:21000] > select \* from stu\_par where month = '201811';

### 5.4.4 增加多个分区

[hadoop103:21000] > alter table stu\_par add partition (month='201812') partition (month='201813');

### 5.4.5 删除分区

[hadoop103:21000] > alter table stu\_par drop partition (month='201812');

### 5.4.5查看分区

[hadoop103:21000] > show partitions stu\_par;

1. DML数据操作

## 6.1 数据导入（基本同hive类似）

注意：impala不支持load data local inpath…

## 6.2 数据的导出

1.impala不支持insert overwrite…语法导出数据

2.impala 数据导出一般使用 impala -o

[root@hadoop103 ~]# impala-shell -q 'select \* from student' -B --output\_delimiter="\t" -o output.txt

[root@hadoop103 ~]# cat output.txt

1001 tignitgn

1002 yuanyuan

1003 haohao

1004 yunyun

Impala 不支持export和import命令

1. 查询
2. 基本的语法跟hive的查询语句大体一样
3. Impala不支持CLUSTER BY, DISTRIBUTE BY, SORT BY
4. Impala中不支持分桶表
5. Impala不支持COLLECT\_SET(col)和explode（col）函数
6. Impala支持开窗函数

[hadoop103:21000] > select name,orderdate,cost,sum(cost) over(partition by month(orderdate)) from business;

1. 函数

## 8.1 自定义函数

1．创建一个Maven工程Hive

2．导入依赖

|  |
| --- |
| <dependencies>  <!-- https://mvnrepository.com/artifact/org.apache.hive/hive-exec -->  <dependency>  <groupId>org.apache.hive</groupId>  <artifactId>hive-exec</artifactId>  <version>1.2.1</version>  </dependency>  </dependencies> |

3．创建一个类

|  |
| --- |
| package com.atguigu.hive;  import org.apache.hadoop.hive.ql.exec.UDF;  public class Lower extends UDF {  public String evaluate (final String s) {    if (s == null) {  return null;  }    return s.toLowerCase();  }  } |

4．打成jar包上传到服务器/opt/module/jars/udf.jar

5. 将jar包上传到hdfs的指定目录

hadoop fs -put hive\_udf-0.0.1-SNAPSHOT.jar /

6. 创建函数

[hadoop103:21000] > create function mylower(string) returns string location '/hive\_udf-0.0.1-SNAPSHOT.jar' symbol='com.atguigu.hive\_udf.Hive\_UDF';

7. 使用自定义函数

[hadoop103:21000] > select ename, mylower(ename) from emp;

8.通过show functions查看自定义的函数

[hadoop103:21000] > show functions;

Query: show functions

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

| return type | signature | binary type | is persistent |

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

| STRING | mylower(STRING) | JAVA | false |

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

1. 存储和压缩

|  |  |  |  |
| --- | --- | --- | --- |
| 文件格式 | 压缩编码 | Impala是否可直接创建 | 是否可直接插入 |
| Parquet | Snappy（默认）, GZIP; | Yes | 支持：CREATE TABLE, INSERT, 查询 |
| Text | LZO，gzip，bzip2，snappy | Yes. 不指定 STORED AS 子句的 CREATE TABLE 语句，默认的文件格式就是未压缩文本 | 支持：CREATE TABLE, INSERT, 查询。如果使用 LZO 压缩，则必须在 Hive 中创建表和加载数据 |
| RCFile | Snappy, GZIP, deflate, BZIP2 | Yes. | 仅支持查询，在 Hive 中加载数据 |
| SequenceFile | Snappy, GZIP, deflate, BZIP2 | Yes. | 仅支持查询，在 Hive 中加载数据 |

注：impala不支持ORC格式

1.创建parquet格式的表并插入数据进行查询

[hadoop104:21000] > create table student2(id int, name string)

> row format delimited

> fields terminated by '\t'

> stored as PARQUET;

[hadoop104:21000] > insert into table student2 values(1001,'zhangsan');

[hadoop104:21000] > select \* from student2;

2.创建sequenceFile格式的表，插入数据时报错

[hadoop104:21000] > insert into table student3 values(1001,'zhangsan');

Query: insert into table student3 values(1001,'zhangsan')

Query submitted at: 2018-10-25 20:59:31 (Coordinator: http://hadoop104:25000)

Query progress can be monitored at: http://hadoop104:25000/query\_plan?query\_id=da4c59eb23481bdc:26f012ca00000000

WARNINGS: Writing to table format SEQUENCE\_FILE is not supported. Use query option ALLOW\_UNSUPPORTED\_FORMATS to override.

1. 优化
2. 尽量将StateStore和Catalog单独部署到同一个节点，保证他们正常通行。
3. 通过对Impala Daemon内存限制（默认256M）及StateStore工作线程数，来提高Impala的执行效率。
4. SQL优化，使用之前调用执行计划
5. 选择合适的文件格式进行存储，提高查询效率。
6. 避免产生很多小文件（如果有其他程序产生的小文件，可以使用中间表，将小文件数据存放到中间表。然后通过insert…select…方式中间表的数据插入到最终表中）
7. 使用合适的分区技术，根据分区粒度测算
8. 使用compute stats进行表信息搜集，当一个内容表或分区明显变化，重新计算统计相关数据表或分区。因为行和不同值的数量差异可能导致impala选择不同的连接顺序时，表中使用的查询。

[hadoop104:21000] > compute stats student;

Query: compute stats student

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

| summary |

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

| Updated 1 partition(s) and 2 column(s). |

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

1. 网络io的优化：

      –a.避免把整个数据发送到客户端

      –b.尽可能的做条件过滤

      –c.使用limit字句

–d.输出文件时，避免使用美化输出

–e.尽量少用全量元数据的刷新

1. 使用profile输出底层信息计划，在做相应环境优化