Presto 作业

*HyperLogLog*算法在*Presto*的应用

HyperLogLog算法原理

HyperLogLog算法来源于论文《HyperLogLog the analysis of a near-optimal cardinality estimation algorithm》。其经常在数据库中被用来统计某一字段的Distinct Value（下文简称DV），比如Redis的HyperLogLog结构。简单来说，它使用固定大小的字节计算任意大小的DV。

基数的概念

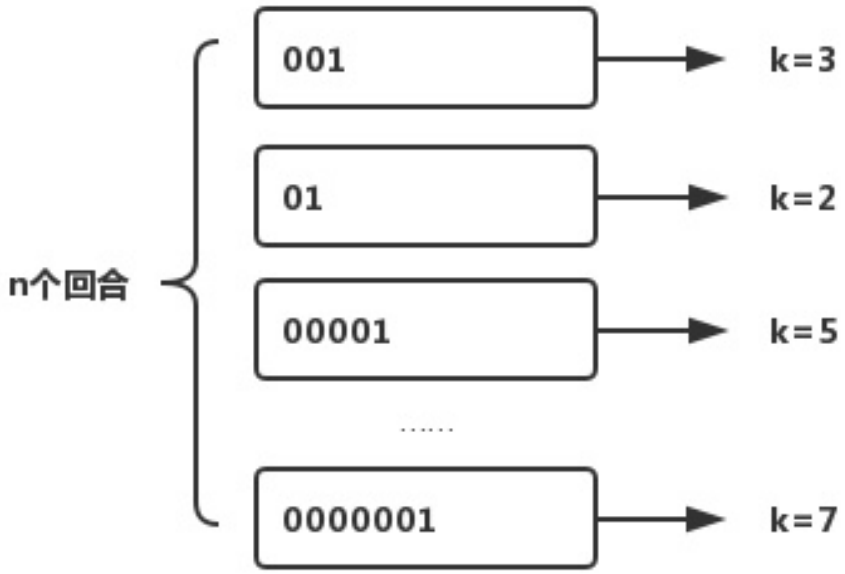
基数就是指一个集合中不同值的数目，比如[a,b,c,d]的基数就是4，[a,b,c,d,a]的基数还是4，因为a重复了一个，不算。基数也可以称之为Distinct Value，简称DV。HyperLogLog算法就是用来计算基数的。

Hyperloglog原理

HyperLogLog实际上不会存储每个元素的值，它使用的是概率算法，通过存储元素的hash值的第一个1的位置，来计算元素数量。

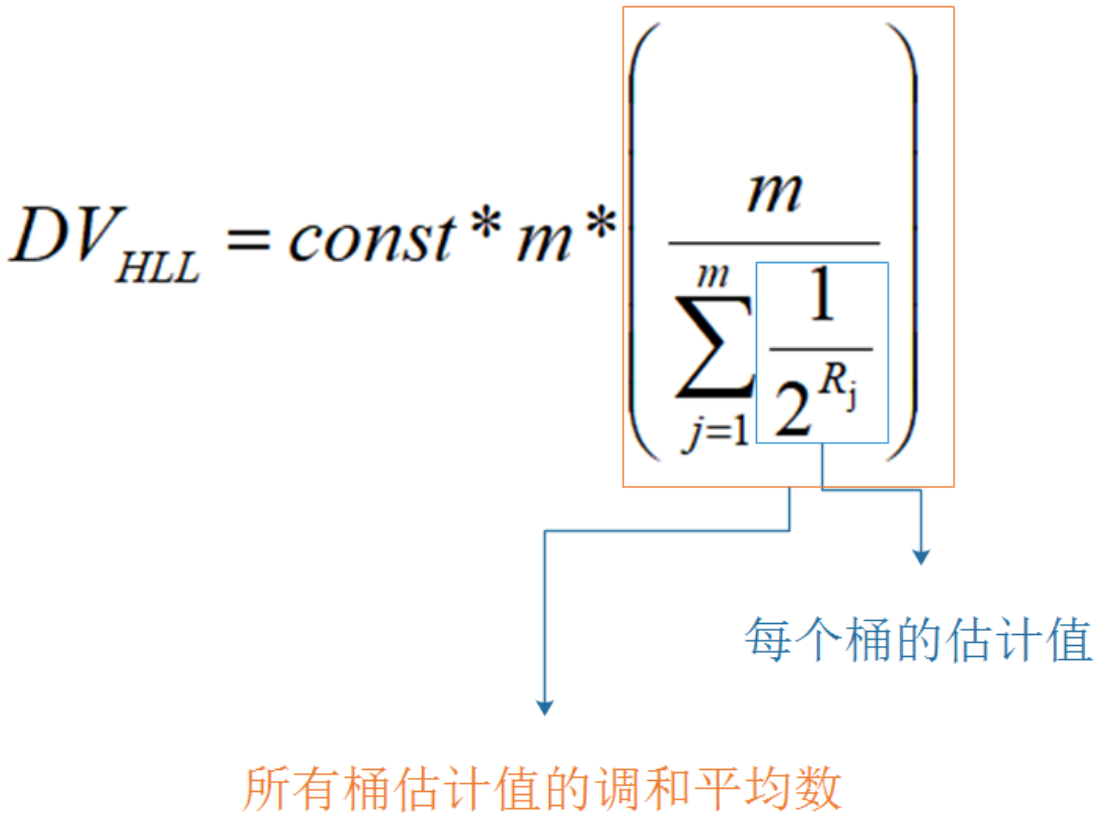
伯努利实验

有一天Jack和丫丫玩抛硬币的游戏，规则是丫丫负责抛硬币，每次抛到正面为一回合，丫丫可以自己决定进行几个回合。最后需要告诉Jack最长的那个回合抛了多少次，再由Jack来猜丫丫一共进行了几个回合。Jack心想：这可不好猜啊，我得算算概率了。于是在脑海中绘制这样一张图。



k是每回合抛到1所用的次数，我们已知的是最大的k值，可以用kmax表示，由于每次抛硬币的结果只有0和1两种情况，因此，kmax在任意回合出现的概率即为(1/2)^k*max，因此可以推测 n=2^k*max 。概率学把这种问题叫做伯努利实验。此时丫丫已经完成了n个回合，并且告诉Jack最长的一次抛了3次，Jack此时也胸有成竹，马上说出他的答案8，最后的结果是：丫丫只抛了一回合，Jack输了，要负责刷碗一个月。

要让这个算法更加准确，于是引入了桶的概念，计算m个桶的加权平均值，这样就能得到比较准确的答案了（实际上还要进行其他修正）。最终的公式为：



其中m是桶的数量，const是修正常数，它的取值会根据m而变化。p=log2*m

```
switch (p) {
   case 4:
       constant = 0.673 * m * m;
   case 5:
       constant = 0.697 * m * m;
   case 6:
       constant = 0.709 * m * m;
   default:
       constant = (0.7213 / (1 + 1.079 / m)) * m * m;
}
```

对于一个输入的字符串，首先得到64位的hash值，用前14位来定位桶的位置（共有2^14，即16384个桶）。后面50位即为伯努利过程，每个桶有6bit，记录第一次出现1的位置count，如果count>oldcount，就用count替换oldcount。

HyperLogLog算法的应用场景

1. Facebook在Presto中使用HyperLogLog（HLL）进行计算密集型操作，例如估算大型数据集中的不同值。通过这种实现，他们能够在处理计数不同的问题时实现高达1000倍的速度提升。
2. 在redis中的应用hyperloglog算法。主要流程：
   - 先对ele求hash（使用的是一种叫做MurMurHash的算法）
   - 将hash的低14位(因为总共有2的14次方个桶)作为桶的编号，选桶，记桶中当前的值为count
   - 从的hash的第15位开始数0，假设从第15位开始有n个连续的0（即前导0）
   - 如果n大于count，则把选中的桶的值置为n，否则不变

   在 Redis 里面，每个 HyperLogLog 键只需要花费 12 KB 内存，就可以计算接近 2^64 个不同元素的基数。
3. 基数统计的一般使用：
   - 统计注册 IP 数
   - 统计每日访问 IP 数
   - 统计页面实时 UV 数
   - 统计在线用户数
   - 统计用户每天搜索不同词条的个数
4. 在流式计算框架（如stream-lib）中实现大数据估值算法，如hyperloglog算法。https://github.com/addthis/stream-lib/blob/5a3bc87c5314f7771ea3968e9015a3d25536343e/src/main/java/com/clearspring/analytics/stream/cardinality/HyperLogLog.java
5. HyperLogLog 算法在监控场景中的运用：在 metric 基数统计场景中，使用 HLL 对时序进行预估，可以保证较低且稳定的内存占用，又可以保证误差率在一个较低的可接受范围内。参考链接：https://www.jianshu.com/p/90766a9cbf24

Hyperloglog算法在presto中进行sql查询应用实现

在presto中实现一个使用hyperloglog算法统计过去一周用户访问量的基数。

1. 先准备数据

```
drop table if exists user_visits_0086;
drop table if exists visit_summaries_0086;

create table user_visits_0086 (
  visit_date date,
  user_id varchar(20)
);

CREATE TABLE visit_summaries_0086 (
  visit_date date,
  hll varbinary
);

insert into user_visits_0086
values(date'2021-09-01','USER_0001'),
(date'2021-09-01','USER_0002'),
(date'2021-09-01','USER_0003'),
(date'2021-09-01','USER_0004'),
(date'2021-09-01','USER_0005'),
(date'2021-09-02','USER_0006'),
(date'2021-09-02','USER_0007'),
(date'2021-09-02','USER_0008'),
(date'2021-09-02','USER_0009'),
(date'2021-09-02','USER_0010'),
(date'2021-09-02','USER_0011'),
(date'2021-09-02','USER_0012'),
(date'2021-09-03','USER_0013'),
(date'2021-09-03','USER_0014'),
(date'2021-09-03','USER_0015'),
(date'2021-09-03','USER_0016'),
(date'2021-09-03','USER_0017'),
(date'2021-09-03','USER_0018'),
(date'2021-09-03','USER_0001'),
(date'2021-09-03','USER_0002'),
(date'2021-09-03','USER_0003'),
```

```
(date'2021-09-04','USER_0004'),
(date'2021-09-04','USER_0005'),
(date'2021-09-04','USER_0006'),
(date'2021-09-04','USER_0007'),
(date'2021-09-04','USER_0008'),
(date'2021-09-04','USER_0009'),
(date'2021-09-04','USER_0010'),
(date'2021-09-05','USER_0011'),
(date'2021-09-05','USER_0012'),
(date'2021-09-05','USER_0013'),
(date'2021-09-05','USER_0014'),
(date'2021-09-05','USER_0015'),
(date'2021-09-05','USER_0016'),
(date'2021-09-05','USER_0017'),
(date'2021-09-05','USER_0018'),
(date'2021-09-05','USER_0019'),
(date'2021-09-05','USER_0020'),
(date'2021-09-05','USER_0021'),
(date'2021-09-05','USER_0022'),
(date'2021-09-05','USER_0023'),
(date'2021-09-06','USER_0024'),
(date'2021-09-06','USER_0025'),
(date'2021-09-06','USER_0026'),
(date'2021-09-06','USER_0027'),
(date'2021-09-06','USER_0028'),
(date'2021-09-06','USER_0029'),
(date'2021-09-06','USER_0030'),
(date'2021-09-07','USER_0031'),
(date'2021-09-07','USER_0032'),
(date'2021-09-07','USER_0033'),
(date'2021-09-07','USER_0034'),
(date'2021-09-07','USER_0035'),
(date'2021-09-07','USER_0036'),
(date'2021-09-07','USER_0037'),
(date'2021-09-07','USER_0038');
```

2. 实现hll查询

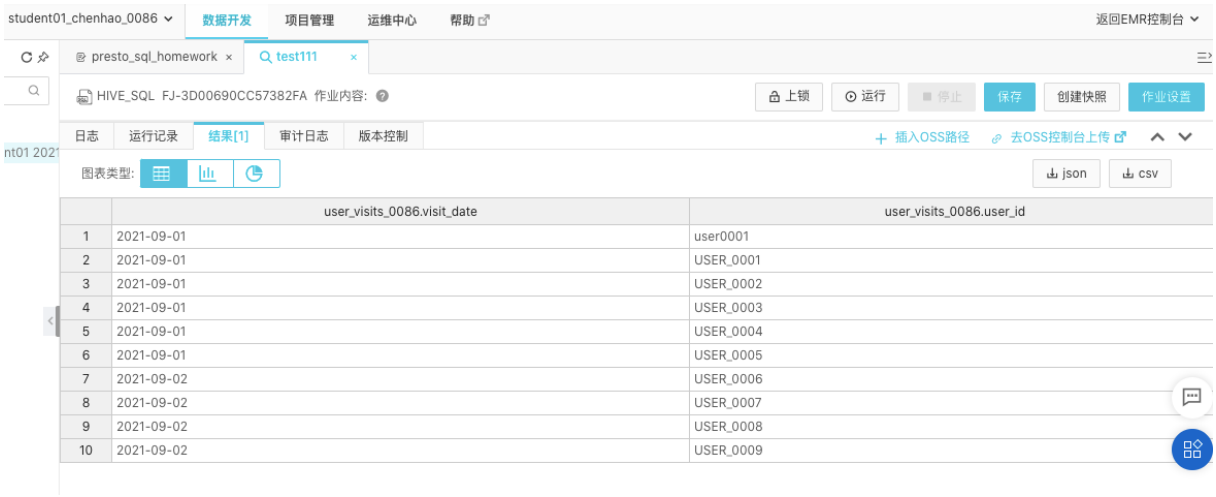
```
INSERT INTO visit_summaries_0086
SELECT visit_date, cast(approx_set(user_id) AS varbinary)
```

```
FROM user_visits_0086
GROUP BY visit_date;

SELECT cardinality(merge(cast(hll AS HyperLogLog))) AS
weekly_unique_users
FROM visit_summaries_0086
WHERE visit_date >= current_date - interval '7' day;
```

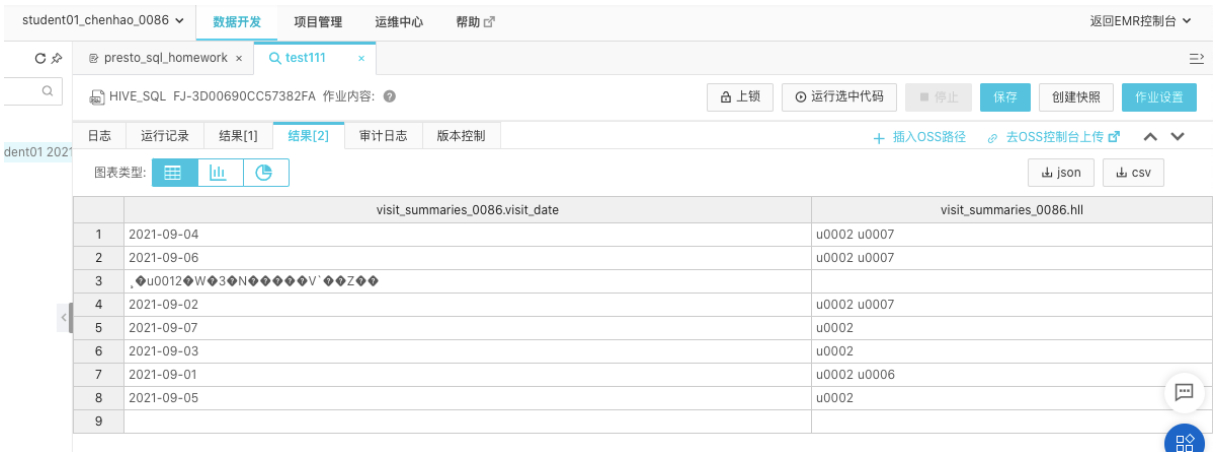
3. 执行过程和结果截图

初始化表user_visits_0086。



	user_visits_0086.visit_date	user_visits_0086.user_id
1	2021-09-01	user0001
2	2021-09-01	USER_0001
3	2021-09-01	USER_0002
4	2021-09-01	USER_0003
5	2021-09-01	USER_0004
6	2021-09-01	USER_0005
7	2021-09-02	USER_0006
8	2021-09-02	USER_0007
9	2021-09-02	USER_0008
10	2021-09-02	USER_0009

表visit_summaries_0086的结果



	visit_summaries_0086.visit_date	visit_summaries_0086.hll
1	2021-09-04	u0002 u0007
2	2021-09-06	u0002 u0007
3	,◆u0012◆W◆3◆N◆◆◆◆◆V`◆◆Z◆◆	
4	2021-09-02	u0002 u0007
5	2021-09-07	u0002
6	2021-09-03	u0002
7	2021-09-01	u0002 u0006
8	2021-09-05	u0002
9		

最后hll算出来的基数结果


student01_chenhao_0086
数据开发
项目管理
运维中心
帮助
返回EMR控制台
test111
spark_sql_for_test
presto_sql_homework
HIVE_SQL FJ-3D00690CC57382FA 作业内容:
上锁
运行
停止
保存
创建快照
作业设置
日志
运行记录
结果[1]
审计日志
版本控制
插入OSS路径
去OSS控制台上传
图表类型:
json
csv
hll_rs2_0086.weekly_unique_users
1
39