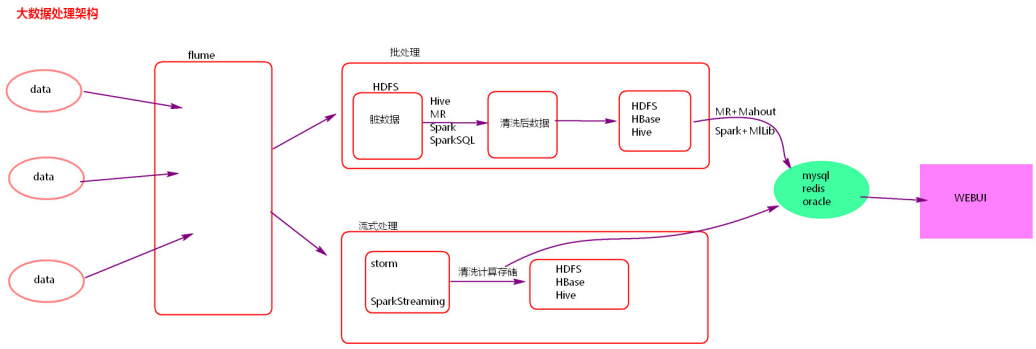
智慧交通项目分析

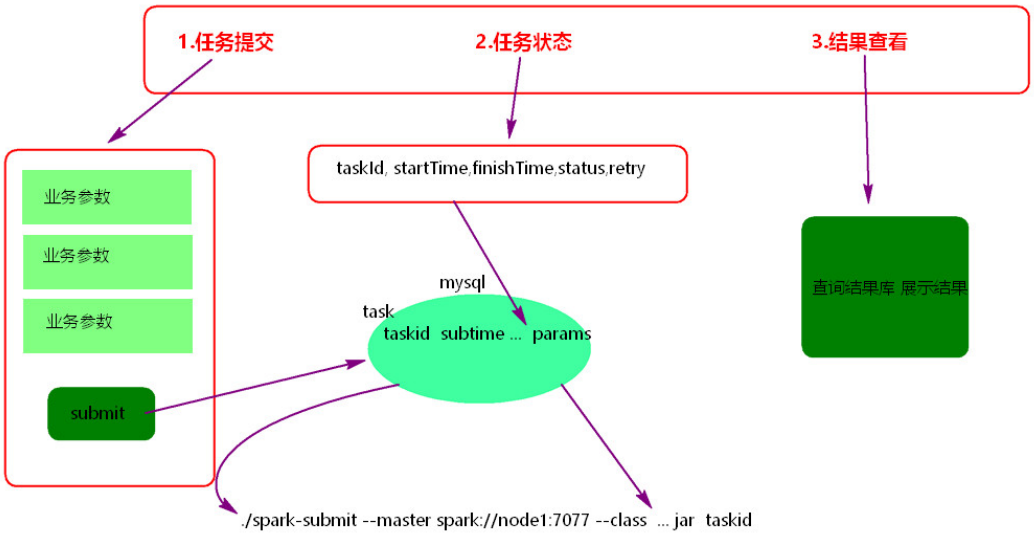
1.数据处理流程



卡口--摄像头：1对多关系

卡口监控
轨迹分析：车的行车轨迹
跟车分析：轨迹相似
碰撞分析：两个区域相同的车
频次分析：卡口下警经过多少车
落脚点分析：长时间停留的地方
昼伏夜出：白天不出现，晚上出现
套牌分析：

过程



1.卡扣监控

【累加器】

正常的卡扣个数，异常的卡扣个数，正常的摄像头个数，异常的摄像头个数，异常的摄像头详细信息

正常卡扣个数：

monitor_camera_info 基本关系表中卡扣与摄像头的关系与在 monitor_flow_action 监控数据表 中，卡扣与摄像头的关系完全对应上

0001:11111,22222

0001 11111 xxx

0001 22222 xxx

异常的卡扣个数：

1.monitor_camera_info 基本关系表中 卡扣 与摄像头的关系，在监控的数据表中 一条都没有对应。

0001:11111,22222 XXXXXXX

2.monitor_camera_info 基本关系表中 卡扣 与摄像头的关系，在监控的数据表中 部分数据有对应。

0001:11111,22222 0001:11111

正常的摄像头个数：

异常的摄像头个数：
异常的摄像头详细信息：0001:11111,22222,33333

''~0004:76789,27449,87911,61106,45624,37726,09506
~0001:70037,23828,34361,92206,76657,26608
~0003:36687,99260,49613,97165
~0006:82302,11645,73565,36440
~0002:60478,07738,53139,75127,16494,48312
~0008:34144,27504,83395,62222,49656,18640

~0007:19179,72906,55656,60720,74161,85939,51743,40565,13972,79216,35128,27369,84616,09553
~0000:67157,85327,08658,57407,64297,15568,31898,36621
~0005:09761,12853,91031,33015,52841,15425,45548,36528
注意：

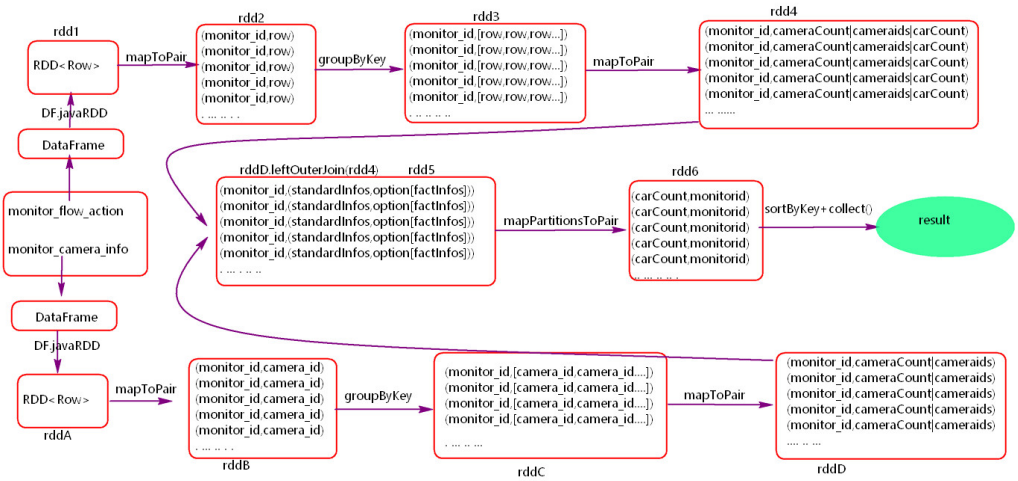
更新累加器与take使用时，take算子可以触发多个job执行，可以造成累加器重复计算。

./spark-submit --master spark://node1:7077,node2:7077 --jars ../lib/fastjson-1.2.11.jar,../lib/mysql-connector-java-5.1.6.jar --class com.bjsxt.spark.skynet.MonitorFlowAnalyze ../lib/Test.jar 1

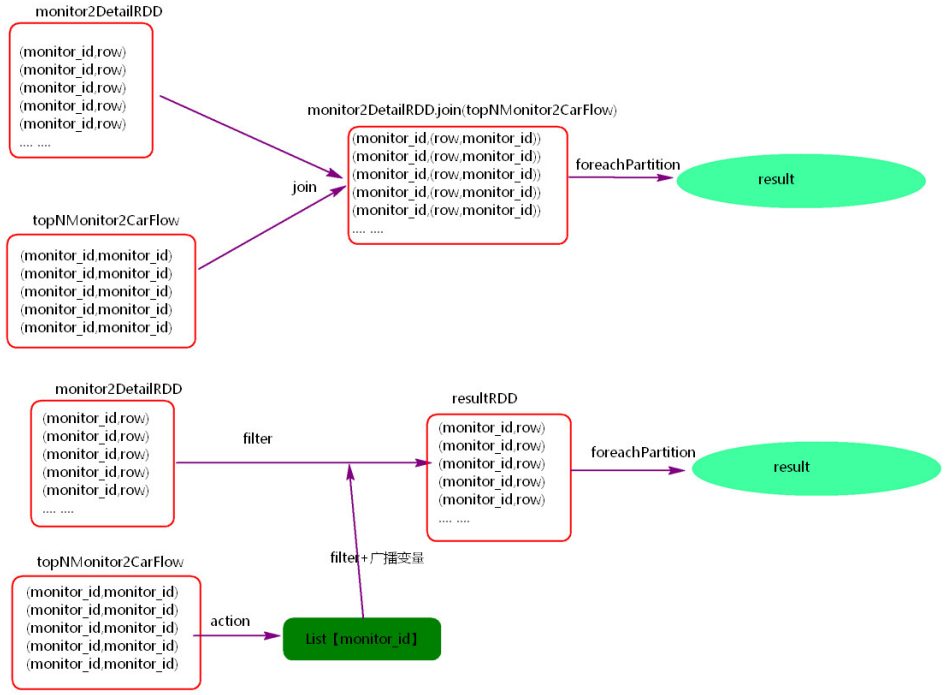
~0001:13846,54785,51995,64341,45994,32228,82054,87746

~0003:38780,08844,03281,07183,50318,87000,16722,11604,26508,45523,46380
~0007:61833,19140,38387
~0005:63920,23464,37389,01219,96765,24844,32101,24141~
~0004:60778,35444,35403,68811,73819,81893
~0006:09621,67028,96375,60036,91237,53743,10305
~0002:24694,01172,25945,79625,83215,72235,26855
~0008:24630,40432,96808,78708,28294

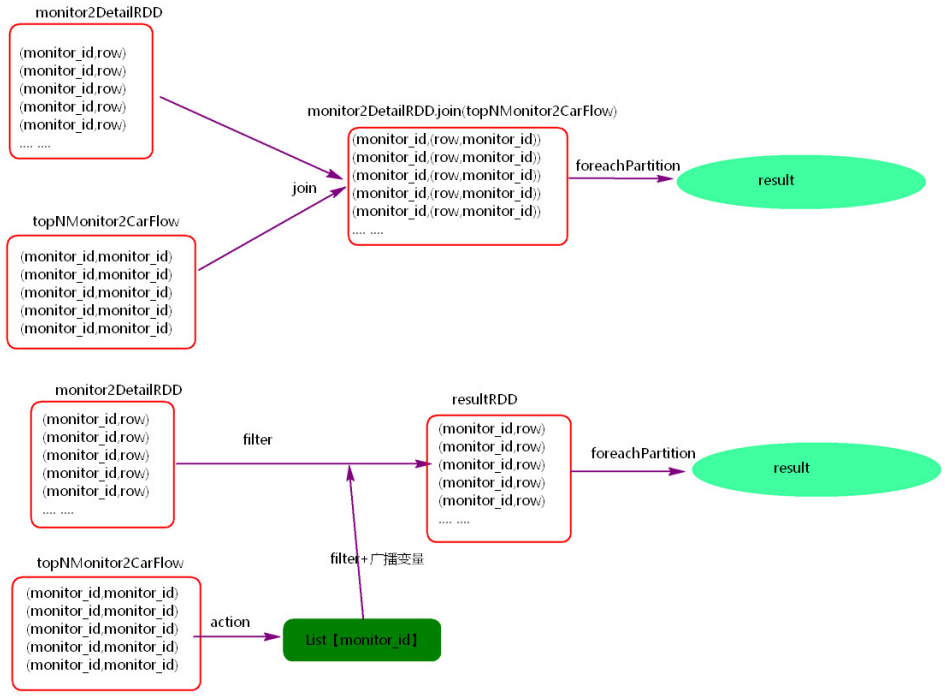
~0000:68070,12865,49505,26035,36931,38053,91868



2.通过车辆数最多的topN卡扣



3.统计topN卡扣下经过的所有车辆详细信息

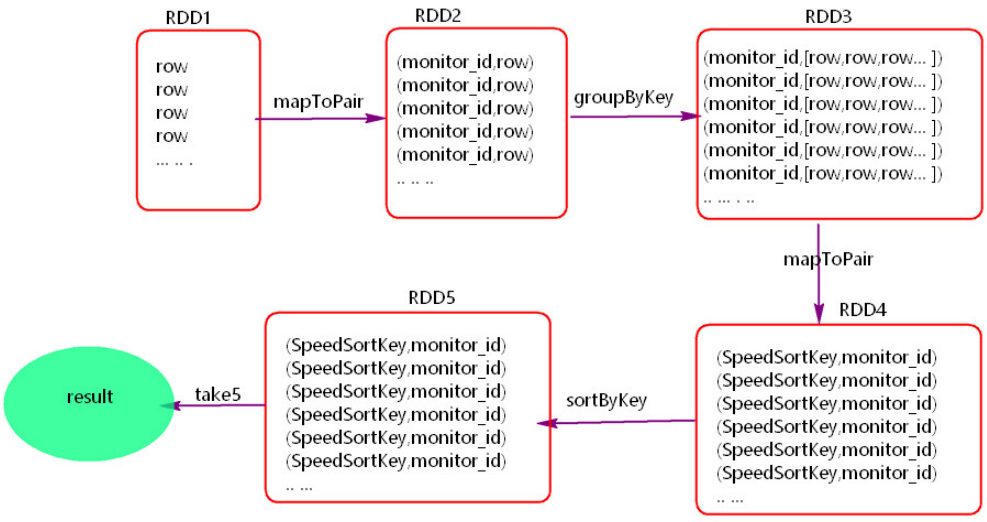


4.车辆通过速度相对比较快的topN卡扣

车速：

120=<speed	高速
90<=speed<120	中速
60<=speed<90	正常
0<speed<60	低速

车辆高速通过的TOPN卡扣

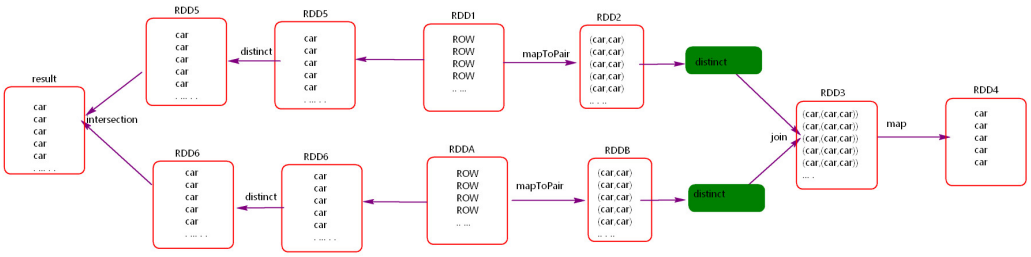


5.卡扣“0001”下所有车辆轨迹

1.过滤日期范围内 卡扣“0001”下 有哪些车辆?

2.过滤日期范围内 这些车辆经过卡扣的时间，按照时间升序排序

6.车辆碰撞



7.随机抽取车辆

在一天中要随机抽取100辆车，抽取的车辆可以权威代表当天交通运行情况。

假如这天一共有10000辆车，要随机抽取100辆车：

sample(true,0.1,seed)

00~01 100　100/10000*100 = 1

01~02 100　1

02~03 100　1

04~05 200　2

05~06 200　2

06~07 300　3

08~09 500　5

09~10 200　2

10~11 200　2

11~12 300　3

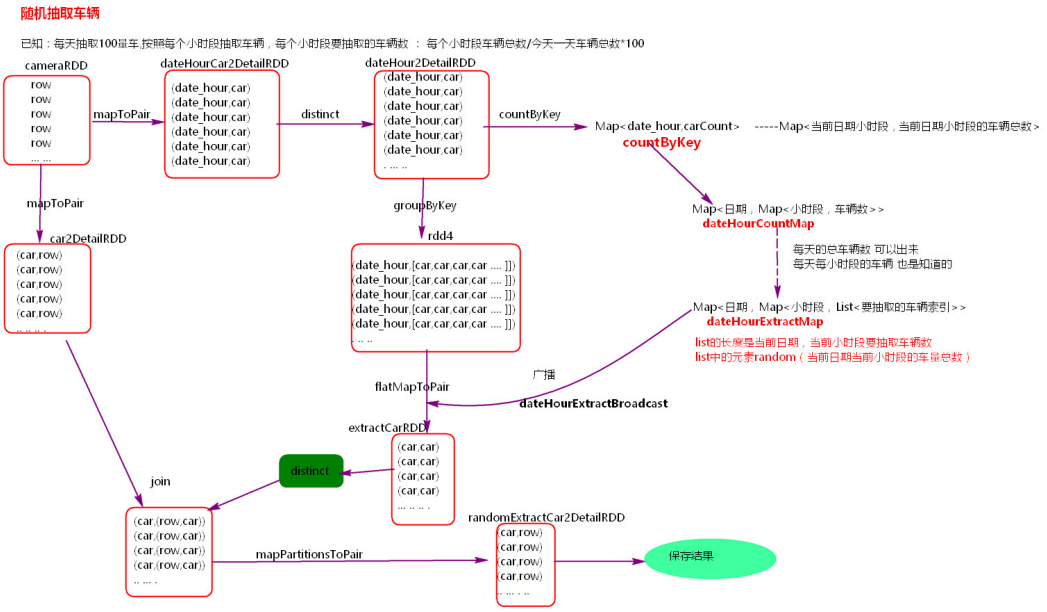
12~13 500　5

13~14 700　7

。。

。。

。 。



8.卡扣流量转换率

一辆车的轨迹：

0001->0002->0003->0001->0002->0004->0005->0001

0001,0002----卡扣0001到卡扣0002 的车流量转化率：通过卡扣0001又通过卡扣0002的次数/通过卡扣0001的次数 2/3

0001,0002,0003 ---- 卡扣0001,0002到0003的车辆转换率：通过卡扣0001,0002,0003的次数 /通过卡扣0001,0002

0001,0002,0003,0004 -----卡扣0001,0002，0003到0004的车辆转换率：通过卡扣0001,0002,0003,0004的次数 /通过卡扣0001,0002,0003

0001,0002,0003,0004,0005 -----卡扣0001,0002，0003,0004到0005的车辆转换率：通过卡扣0001,0002,0003,0004,0005的次数 /通过卡扣0001,0002,0003,0004的次数

手动输入卡扣号：

0001,0002,0003,0004,0005

求：

0001,0002

0001,0002,0003

0001,0002,0003,0004

0001,0002,0003,0004,0005

粤A11111：

("0001"，100)

("0001,0002",30)

("0001,0002,0003",10)

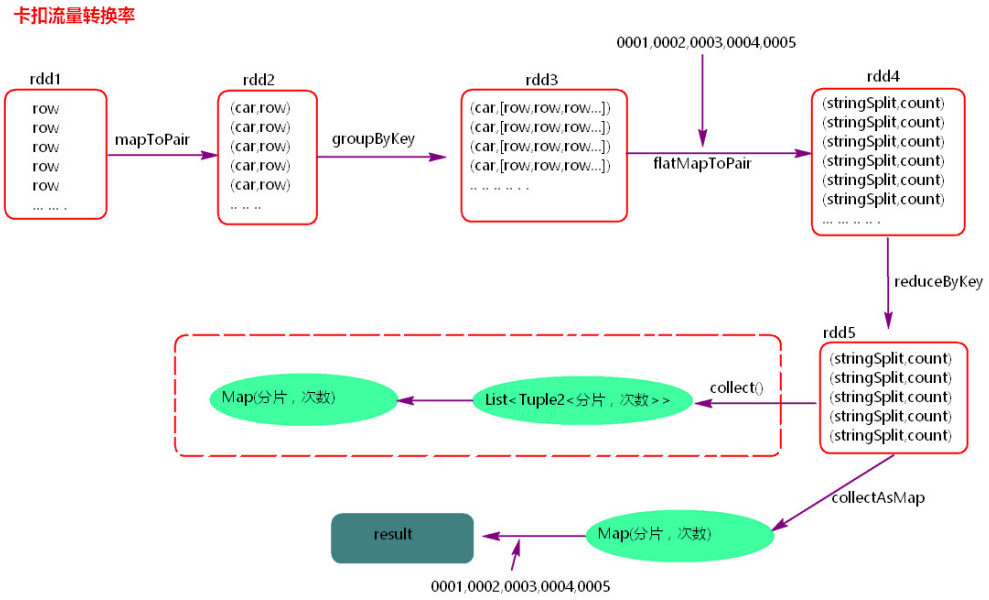
粤B22222：

("0001"，200)

("0001,0002",100)

("0001,0002,0003",70)

("0001,0002,0003,0004",10)



9.实时道路拥堵情况

计算一段时间内卡扣下通过的车辆的平均速度。

这段时间不能太短，也不能太长。就计算当前时间的前五分钟 当前卡扣下通过所有车辆的平均速度。

每隔5s 计算一次当前卡扣过去5分钟 所有车辆的平均速度。

SparkStreaming 窗口函数

window lenth:5min

slide interval:5s

10.动态改变广播变量

11.统计每个区域中车辆最多的前3道路

道路车辆：道路中的每个卡扣经过的车辆累加

天河区 元岗路1 0001=30,0002=50,0003=100,0004=20 200
天河区 元岗路2 0005=50,0006=100 150
天河区 元岗路3 100
越秀区 xxx1 200
越秀区 xxx2 150
越秀区 xxx3 100

Hive 表 --t1 :

monitor_id car road_id area_id

---

areaId area_name road_id monitor_id car ------ tmp_car_flow_basic

sql:

select area_name,road_id,count(car) as car_count,UDAF(monitor_id) as monitor_infos from t1 group by area_name,road_id ---- tmp_area_road_flow_count

开窗函数：row_number() over (partition by xxx order by xxx) rank

select area_name,road_id,car_count,monitor_infos, row_number() over (partition by area_name order by car_count desc) rank from tmp_area_road_flow_count ---- tmp

select area_name,road_id,car_count,monitor_infos from tmp where rank <=3

```
-----------------------------------------------------------------------
总sql:
        select
            area_name,road_id,car_count,monitor_infos
        from
            (
            select
                area_name,road_id,car_count,monitor_infos, row_number()
over (partition by area_id order by carCount desc ) rank
            from
                (
                    select
                        area_name,road_id,count(car) as car_count
,UDAF(monitor_id) as monitor_infos
                    from
                        t1
                    group by area_name,road_id
                ) t2
            ) t3
        where rank <=3

=========================================================
=========================================
sql:
        select prefix_area_name_road_id,count(car) as
car_count,UDAF(monitor_id) as monitor_infos from t1 group by
prefix_area_name_road_id          ---- tmp_area_road_flow_count

        select area_name,road_id,car_count,monitor_infos, row_number()
over (partition by area_name order by car_count desc ) rank from
tmp_area_road_flow_count  ---- tmp

        select area_name,road_id,car_count,monitor_infos from tmp where
```

rank <=3

总sql：

```
select
    area_name,road_id,car_count,monitor_infos
from
    (
    select
        area_name,road_id,car_count,monitor_infos, row_number()
over (partition by area_id order by carCount desc ) rank
    from
        (
            select
                area_name,road_id,count(car) as car_count
,UDAF(monitor_id) as monitor_infos
            from
                t1
            group by area_name,road_id
        ) t2
    ) t3
where rank <=3
```