# 浙江大华车辆大数据ADS联合测试方案

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **修订版本** | **修订日期** | **修订内容** | **修订人** |
| 1.0 | 2017-05-20 | 初稿 | 李琦 |
|  |  |  |  |
|  |  |  |  |

**目录**

[浙江大华车辆大数据ADS联合测试方案 1](#_Toc483033992)

[一、测试项目 4](#_Toc483033993)

[1、测试需求分析 4](#_Toc483033994)

[2、ADS平台介绍 4](#_Toc483033995)

[3、浙江大华车辆大数据业务需求分析 5](#_Toc483033996)

[二、测试设计 5](#_Toc483033997)

[1、测试环境 5](#_Toc483033998)

[2、测试组网分析 5](#_Toc483033999)

[3、测试数据准备 6](#_Toc483034000)

[3.1 准备java环境 6](#_Toc483034001)

[3.2用Python脚本生成测试数据 6](#_Toc483034002)

[3.3 工具导入数据 6](#_Toc483034003)

[4、详细测试方法 6](#_Toc483034004)

[4.1 测试工具 6](#_Toc483034005)

[4.2 表结构 7](#_Toc483034006)

[5、测试目标 8](#_Toc483034007)

[三、测试用例 8](#_Toc483034008)

[1、时间+地点碰撞 8](#_Toc483034009)

[2、首次出现 9](#_Toc483034010)

[3、频繁出现 10](#_Toc483034011)

[4、同行车分析 11](#_Toc483034012)

[5、相似车牌串并 11](#_Toc483034013)

# 一、测试项目

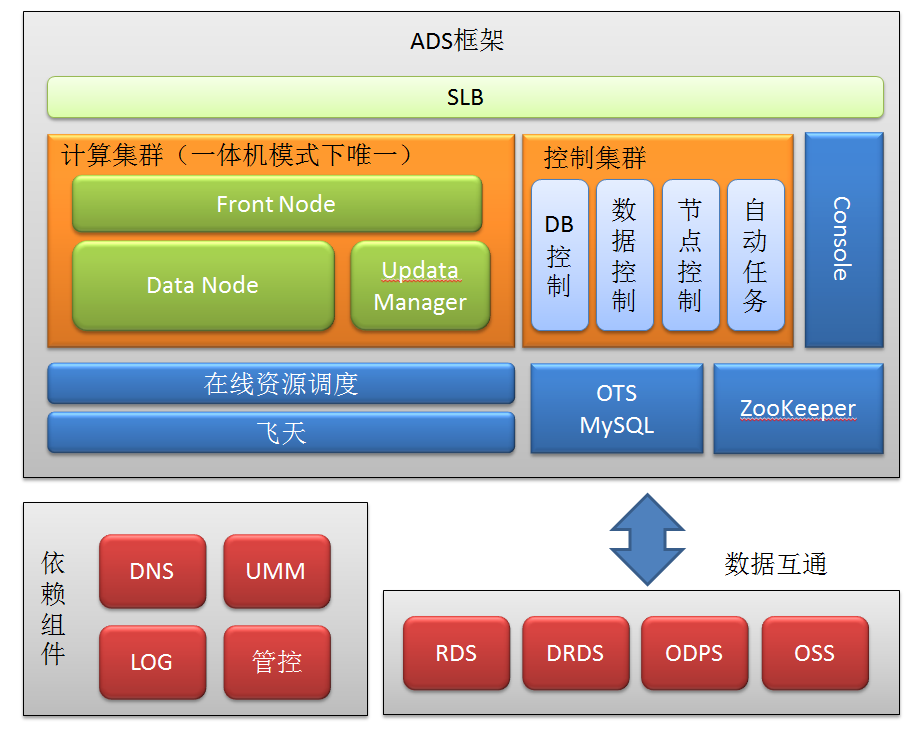
## 1、测试需求分析

浙江大华的车辆大数据研判系统是通过先进的大数据分析技术，以过车数据和车辆图片为主线，通过对车辆图片进行二次识别获取结构化信息，并联动公安网上业务数据（如人口库、车辆登记库、盗抢车辆库、前科犯罪人员库等）进行车辆、人员、案件、物品、现场勘查、电磁轨迹及社会资源数据的关联串并和碰撞比对。随着大数据技术的迅猛发展，浙江大华作为该行业领先的服务提供商，居安思危，希望找到更高性能的实时多维分析型数据库为其提供优越的查询能力，提升行业竞争力。ADS作为一款针对海量数据的实时高并发在线分析服务，完美的满足这一需求。

## 2、ADS平台介绍

ADS是阿里巴巴集团为了满足淘宝、支付宝等众多业务在线分析、即席查询、多维分析等需求的一款产品。该产品和阿里上层许多应用关系紧密，提供的功能众多。特色功能有四类：全智能索引、多值列、空间检索和海量Dump操作能力。

本质上，ADS是一款强化计算能力的MPP数据库。具备高速计算、查询能力的同时具备扩展性，支持SQL92标准，也方便上层应用使用ADS进行业务的再开发。ADS又不同于标准数据库产品，ADS自身对性能的极限追求导致它在应用时不可能随心所欲地任由用户随意修改和划分。因此，区别于售卖“空间”的普通数据库产品，ADS更多地是提供一种服务“能力”，用来帮助用户处理自己手头的数据。



ADS系统结构图

## 3、浙江大华车辆大数据业务需求分析

1）时间+地点碰撞

通过时间和卡口两个信息作为一个条件对，通过取交集筛选出在一个时间多个地点出现和一个地点多个时间经过的车辆信息。

2）首次出现

筛选出在设定的回溯天数内首次经过所选卡口的车辆信息。

3）频繁出现

筛选出在设定的时间内在某一地点（卡口）反复出现达到设定次数的车辆信息。

4）同行车分析

通过设置被尾随车辆的车牌号码、经过时间段等信息，分析出尾随车辆信息。

5）相似车牌串并

根据已输入车牌号码自动串并，分析出车牌号码相差一位或者多位的相似车辆信息。

# 二、测试设计

## 1、测试环境

测试过程中，需要的硬件、软件如下表格：

表1-硬件规格

|  |  |  |  |
| --- | --- | --- | --- |
| 服务器机型 | 详细配置 | | 数量 |
|  | CPU |  |  |
|  | 内存 |  |  |
|  | 硬盘 |  |  |
|  | 网卡 |  |  |

表2-软件环境

|  |  |
| --- | --- |
| 软件名称 | 版本号 |
| OS |  |
| ADS |  |
| 车辆大数据平台 |  |
|  |  |
|  |  |

## 2、测试组网分析

## 3、测试数据准备

### 3.1 准备java环境

1. 准备一台高配服务器，供快速生成数据使用；
2. OS：建议安装alios，便于安装盘古客户端，使用工具往盘古上传数据；
3. Java环境：jdk1.8
4. 运行脚本，生成数据(27亿)

### 3.2用Python脚本生成测试数据

1. 准备一台服务器；
2. 服务器安装OS，安装python；
3. 运行脚本，生成数据；

### 3.3 工具导入数据

#### 3.3.1 XX工具导入数据

1. 适合小表导入，操作步骤相对简单；
2. 对数据格式有一定要求，导入过程中出现错误，会中断；

#### 3.3.2XX工具导入数据

1. 适用大量数据导入，步骤相对复杂；
2. 数据导入过程中出现错误，会跳过继续进行导入；

## 4、详细测试方法

### 4.1 测试工具

1. 数据生成工具：python脚本
2. 数据导入工具：
3. 并发测试工具：Jmeter或LR

### 4.2 表结构

表一：

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | C\_PICRECORD：卡口过车信息表 | | |
| 序号 | 字段名 | 类型 | 描述 |
| 1 | ID | NUMBER(11) | 主键 |
| 2 | DEV\_ID | VARCHAR2(50) | 设备ID |
| 3 | DEV\_CHNNUM | NUMBER(5) | 通道序号 |
| 4 | DEV\_NAME | VARCHAR2(50) | 设备名称 |
| 5 | DEV\_CHNNAME | VARCHAR2(50) | 通道名称 |
| 6 | CAR\_NUM | VARCHAR2(12) | 车牌号 |
| 7 | CAR\_NUMTYPE | NUMBER(3) | 车牌类型 |
| 8 | CAR\_NUMCOLOR | NUMBER(3) | 车牌颜色 |
| 9 | CAR\_SPEED | NUMBER(3) | 车速 |
| 10 | CAR\_TYPE | NUMBER(3) | 车型 |
| 11 | CAR\_COLOR | NUMBER(3) | 车身颜色 |
| 12 | CAR\_LENGTH | NUMBER(4) | 车长 |
| 13 | CAR\_DIRECT | VARCHAR2(8) | 行车方向 |
| 14 | CAR\_WAY\_CODE | VARCHAR2(8) | 车道编号 |
| 15 | CAP\_TIME | NUMBER(12) | 抓图时间 |
| 16 | CAP\_DATE | DATE | 抓图日期 |
| 17 | INF\_NOTE | VARCHAR2(255) | 备注信息 |
| 18 | MAX\_SPEED | NUMBER(3) | 最高限制速度 |
| 19 | MIN\_SPEED | NUMBER(3) | 最低限制速度 |
| 20 | CAR\_IMG\_URL | VARCHAR2(255) | 图片存放路径 |
| 21 | CAR\_IMG1\_URL | VARCHAR2(255) | 图片存放路径 |
| 22 | CAR\_IMG2\_URL | VARCHAR2(255) | 图片存放路径 |
| 23 | CAR\_IMG3\_URL | VARCHAR2(255) | 图片存放路径 |
| 24 | CAR\_IMG4\_URL | VARCHAR2(255) | 图片存放路径 |
| 25 | CAR\_IMG5\_URL | VARCHAR2(255) | 图片存放路径 |
| 26 | REC\_STAT | NUMBER(2) | 记录状态 |
| 27 | DEV\_CHNID | VARCHAR2(50) | 设备通道ID |
| 28 | CAR\_IMG\_COUNT | NUMBER(1) | 照片数量 |
| 29 | SAVE\_FLAG | NUMBER(1) | 记录图片是否要永久保存 |
| 30 | DC\_CLEANFLAG | NUMBER(3) | DATACLEAN程序使用表示，是否已清除图片 |
| 31 | PIC\_ID | NUMBER(3) | 图片组ID |
| 32 | CAR\_IMG\_PLATE\_TOP | NUMBER(4) | 车牌在图片中的坐标TOP |
| 33 | CAR\_IMG\_PLATE\_LEFT | NUMBER(4) | 车牌在图片中的坐标left |
| 34 | CAR\_IMG\_PLATE\_BOTTOM | NUMBER(4) | 车牌在图片中的坐标bottom |
| 35 | CAR\_IMG\_PLATE\_RIGHT | NUMBER(4) | 车牌在图片中的坐标right |
| 36 | CAR\_BRAND | NUMBER(4) | 车辆商标 |
| 37 | ISSAFETYBELT | NUMBER(2) | 是否系安全带 |
| 38 | ISVISOR | NUMBER(2) | 是否有遮阳板 |
| 39 | BIND\_STAT | NUMBER(2) | 是否是已合成图片 |
| 40 | CAR\_NUM\_PIC | VARCHAR2(255) | 车牌小图片，后续3.0中车牌小图坐标点位数据不再使用，都取此字段组成车牌小图 |
| 41 | COMBINED\_PIC\_URL | VARCHAR2(255) | 合成图片保存路径URL |

表二：

## 5、测试目标

本测试方案旨在浙江大华车辆大数据平台与ADS平台的契合度，重点在于性能测试。对于实际用到的时间+地点碰撞、首次出现、频繁出现、同行车分析、相似车牌串并等功能进行详细的性能测试。

# 三、测试用例

## 1、时间+地点碰撞

|  |  |
| --- | --- |
| 测试目的 | 找出在多个相同时间段内，相同地点出现的车辆 |
| 测试数据要求 | 30亿条 |
| 测试条件 |  |
| 参考说明 | Where时间范围1个月内 |
| 预期结果 | 秒级（5s以内） |
| 测试SQL | SELECT car\_num,  max(cap\_date) AS maxh  FROM C\_PICRECORD  where CAR\_NUM!= '0'  AND DEV\_CHNID IN('1000000$1$0$0', '1000001$1$0$0')  AND CAP\_DATE>= '2015-11-29 04:00:00'  AND CAP\_DATE<= '2015-11-29 15:00:00'  GROUP BY car\_num  UNION ALL SELECT car\_num,  max(cap\_date) AS maxh  FROM C\_PICRECORD  where CAR\_NUM!= '0'  AND DEV\_CHNID IN('1000000$1$0$0', '1000001$1$0$0')  AND CAP\_DATE>= '2015-11-29 04:00:00'  AND CAP\_DATE<= '2015-11-29 15:00:00'  GROUP BY car\_num  UNION ALL SELECT car\_num,  max(cap\_date) AS maxh  FROM C\_PICRECORD  where CAR\_NUM!= '0'  AND DEV\_CHNID IN('1000000$1$0$0', '1000001$1$0$0')  AND CAP\_DATE>= '2015-11-29 04:00:00'  AND CAP\_DATE<= '2015-11-29 15:00:00'  GROUP BY car\_num |
| 实测结果 |  |

## 2、首次出现

|  |  |
| --- | --- |
| 测试目的 | 找出在指定时间段内，且上次出现的时间在回溯天数之前（或从来没出现过）的车辆 |
| 测试数据要求 | 30亿条 |
| 测试条件 | 1个月、3个月、半年都做一次纪录 |
| 参考说明 |  |
| 预期结果 | 1、1个月、3个月秒级（5s以内）；半年秒级（8s以内待参考） |
| 测试SQL | SELECT c.ID ID, c.CAR\_NUM CAR\_NUM, c.CAR\_TYPE CAR\_NUM\_TPYE, c.CAP\_DATE CAP\_DATE, c.DEV\_CHNNAME LOCALE\_NAME, c.DEV\_CHNID LOCALE\_CODE, c.CAR\_IMG\_URL CAR\_IMG\_URL, c.CAR\_NUM\_PIC CAR\_NUM\_PIC, c.CAR\_COLOR  FROM C\_PICRECORD c  left join (SELECT distinct CAR\_NUM  FROM C\_PICRECORD  WHERE CAP\_DATE >= '2015-09-01 00:00:00' - #{hsts}  AND CAP\_DATE < '2015-09-30 00:00:00'  AND c.CAR\_NUM = t2.CAR\_NUM  AND DEV\_CHNID IN ('','','')) t2  on c.car\_num = t2.car\_num  left join  (SELECT car\_num FROM VIAS\_CAR\_INFO where CAR\_STATUS='M') t3 on c.car\_num = t3.car\_num  WHERE t2.car\_num is null and t3.car\_num is null  AND c.CAR\_NUM<> '0'  AND c.CAP\_DATE>= '2015-09-01 00:00:00'  AND c.CAP\_DATE<= '2015-09-30 00:00:00'  AND c.CAR\_NUM LIKE '浙A\_999%'  AND c.DEV\_CHNID IN ('','','')  AND c.CAR\_COLOR = 5  AND c.CAR\_TYPE = 1  AND c.CAR\_NUM NOT LIKE '浙A1999%' |
| 实测结果 |  |

## 3、频繁出现

|  |  |
| --- | --- |
| 测试目的 | 在时间段、选择地点内频繁出现的车辆 |
| 测试数据要求 | 30亿条 |
| 测试条件 | 1个月、3个月、半年都做一次纪录 |
| 参考说明 |  |
| 预期结果 | 1个月、3个月秒级（5s以内）；半年秒级（8s以内待参考） |
| 测试SQL | SELECT APPEAR\_TIMES,car\_num, car\_type FROM (  SELECT COUNT(\*) as APPEAR\_TIMES,CAR\_NUM,CAR\_TYPE  FROM C\_PICRECORD WHERE 1=1  AND CAP\_DATE >= '2015-06-01 00:00:00'  AND CAP\_DATE <= '2015-06-30 24:00:00'  AND CAR\_NUM LIKE '豫A%'  GROUP BY CAR\_TYPE,CAR\_NUM  ) a where a.APPEAR\_TIMES>= 30  ORDER BY APPEAR\_TIMES DESC,CAR\_NUM |
| 实测结果 |  |

## 4、同行车分析

|  |  |
| --- | --- |
| 测试目的 | 查询指定车牌，相同时间（2分钟内），相同地点经过的车辆 |
| 测试数据要求 | 30亿条 |
| 测试条件 | 1个月、3个月、半年都做一次记录 |
| 参考说明 | Date需要计算到分钟级，考虑使用timestamps |
| 预期结果 | 1个月、3个月秒级（3s以内）；半年秒级（5s以内待参考） |
| 测试SQL | select car\_num,  count(\*) as cnt  from C\_PICRECORD  where(dev\_chnid= '4101010005777000003031$1$0$0'  and cap\_date>= '2006-10-11 12:43:19'  and cap\_date<= '2006-10-11 12:47:19')  or(dev\_chnid= '4101010005777000003061$1$0$0'  and cap\_date>= '2011-03-13 18:22:22'  and cap\_date<= '2011-03-13 18:26:22')  or(dev\_chnid= '4101010005777000003080$1$0$0'  and cap\_date>= '2009-07-01 19:35:56'  and cap\_date<= '2009-07-01 19:39:56')  or(dev\_chnid= '4101010005777000003080$1$0$0'  and cap\_date>= '2012-12-06 13:21:15'  and cap\_date<= '2012-12-06 13:25:15')  or(dev\_chnid= '4101010005777000003081$1$0$0'  and cap\_date>= '2013-10-27 13:52:30'  and cap\_date<= '2013-10-27 13:56:30')  group by car\_num  having cnt> 3 |
| 实测结果 |  |

## 5、相似车牌串并

|  |  |
| --- | --- |
| 测试目的 | 查询指定车牌的相似车牌，在该段时间内的过车情况 |
| 测试数据要求 | 30亿条 |
| 测试条件 | 1个月、3个月、半年都做一次纪录 |
| 参考说明 |  |
| 预期结果 | 1、1个月、3个月秒级（5s以内）；半年秒级（8s以内待参考） |
| 测试SQL | select (decode(substr(a.car\_num, 1, 1), substr('${b\_carNum}', 1, 1), substr(a.car\_num, 1, 1), '\*') ||  decode(substr(a.car\_num, 2, 1), substr('${b\_carNum}', 2, 1), substr(a.car\_num, 2, 1), '\*') ||  decode(substr(a.car\_num, 3, 1), substr('${b\_carNum}', 3, 1), substr(a.car\_num, 3, 1), '\*') ||  decode(substr(a.car\_num, 4, 1), substr('${b\_carNum}', 4, 1), substr(a.car\_num, 4, 1), '\*') ||  decode(substr(a.car\_num, 5, 1), substr('${b\_carNum}', 5, 1), substr(a.car\_num, 5, 1), '\*') ||  decode(substr(a.car\_num, 6, 1), substr('${b\_carNum}', 6, 1), substr(a.car\_num, 6, 1), '\*') ||  decode(substr(a.car\_num, 7, 1), substr('${b\_carNum}', 7, 1), substr(a.car\_num, 7, 1), '\*')) carNumStyle, count(1) similarCarNum  FROM (SELECT CAR\_NUM FROM C\_PICRECORD  WHERE 1=1  AND CAP\_DATE >= to\_date('2015-08-08 14:00:00','yyyy-mm-dd hh24:mi:ss')  AND CAP\_DATE <= to\_date('2015-08-08 15:00:00','yyyy-mm-dd hh24:mi:ss')  GROUP BY CAR\_NUM, car\_type) a  where (decode(substr(a.car\_num, 1, 1), substr('${b\_carNum}', 1, 1), 1, 0) +  decode(substr(a.car\_num, 2, 1), substr('${b\_carNum}', 2, 1), 1, 0) +  decode(substr(a.car\_num, 3, 1), substr('${b\_carNum}', 3, 1), 1, 0) +  decode(substr(a.car\_num, 4, 1), substr('${b\_carNum}', 4, 1), 1, 0) +  decode(substr(a.car\_num, 5, 1), substr('${b\_carNum}', 5, 1), 1, 0) +  decode(substr(a.car\_num, 6, 1), substr('${b\_carNum}', 6, 1), 1, 0) +  decode(substr(a.car\_num, 7, 1), substr('${b\_carNum}', 7, 1), 1, 0)) = 7 - ${diffBitNum}  group by  (decode(substr(a.car\_num, 1, 1), substr('${b\_carNum}', 1, 1), substr(a.car\_num, 1, 1), '\*') ||  decode(substr(a.car\_num, 2, 1), substr('${b\_carNum}', 2, 1), substr(a.car\_num, 2, 1), '\*') ||  decode(substr(a.car\_num, 3, 1), substr('${b\_carNum}', 3, 1), substr(a.car\_num, 3, 1), '\*') ||  decode(substr(a.car\_num, 4, 1), substr('${b\_carNum}', 4, 1), substr(a.car\_num, 4, 1), '\*') ||  decode(substr(a.car\_num, 5, 1), substr('${b\_carNum}', 5, 1), substr(a.car\_num, 5, 1), '\*') ||  decode(substr(a.car\_num, 6, 1), substr('${b\_carNum}', 6, 1), substr(a.car\_num, 6, 1), '\*') ||  decode(substr(a.car\_num, 7, 1), substr('${b\_carNum}', 7, 1), substr(a.car\_num, 7, 1), '\*')) |
| 实测结果 |  |