日志

目录

[Overview 9](#_Toc502824029)

[2018 10](#_Toc502824030)

[Week\_01 10](#_Toc502824031)

[2018/1/1 10](#_Toc502824032)

[2018/1/2 10](#_Toc502824033)

[2018/1/3 10](#_Toc502824034)

[2018/1/4 10](#_Toc502824035)

[2018/1/5 10](#_Toc502824036)

[2018/1/6 10](#_Toc502824037)

[2018/1/7 10](#_Toc502824038)

[Week\_02 11](#_Toc502824039)

[Week\_03 11](#_Toc502824040)

[Week\_04 11](#_Toc502824041)

[Week\_05 11](#_Toc502824042)

[Week\_06 11](#_Toc502824043)

[Week\_07 11](#_Toc502824044)

[Week\_08 11](#_Toc502824045)

[Week\_09 11](#_Toc502824046)

[Week\_10 11](#_Toc502824047)

[Week\_11 11](#_Toc502824048)

[Week\_12 11](#_Toc502824049)

[Week\_13 11](#_Toc502824050)

[Week\_14 11](#_Toc502824051)

[Week\_15 11](#_Toc502824052)

[Week\_16 11](#_Toc502824053)

[Week\_17 11](#_Toc502824054)

[Week\_18 12](#_Toc502824055)

[Week\_19 12](#_Toc502824056)

[Week\_20 12](#_Toc502824057)

[Week\_21 12](#_Toc502824058)

[Week\_22 12](#_Toc502824059)

[Week\_23 12](#_Toc502824060)

[Week\_24 12](#_Toc502824061)

[Week\_25 12](#_Toc502824062)

[Week\_26 12](#_Toc502824063)

[Week\_27 12](#_Toc502824064)

[Week\_28 12](#_Toc502824065)

[Week\_29 12](#_Toc502824066)

[Week\_30 12](#_Toc502824067)

[Week\_31 12](#_Toc502824068)

[Week\_32 12](#_Toc502824069)

[Week\_33 12](#_Toc502824070)

[Week\_34 13](#_Toc502824071)

[Week\_35 13](#_Toc502824072)

[Week\_36 13](#_Toc502824073)

[Week\_37 13](#_Toc502824074)

[Week\_38 13](#_Toc502824075)

[Week\_39 13](#_Toc502824076)

[Week\_40 13](#_Toc502824077)

[Week\_41 13](#_Toc502824078)

[Week\_42 13](#_Toc502824079)

[Week\_43 13](#_Toc502824080)

[Week\_44 13](#_Toc502824081)

[Week\_45 13](#_Toc502824082)

[Week\_46 13](#_Toc502824083)

[Week\_47 13](#_Toc502824084)

[Week\_48 13](#_Toc502824085)

[Week\_49 13](#_Toc502824086)

[Week\_50 14](#_Toc502824087)

[Week\_51 14](#_Toc502824088)

[Week\_52 14](#_Toc502824089)

[2017 15](#_Toc502824090)

[Week\_52 15](#_Toc502824091)

[2017/12/25 15](#_Toc502824092)

[2017/12/26 15](#_Toc502824093)

[2017/12/27 15](#_Toc502824094)

[2017/12/28 15](#_Toc502824095)

[2017/12/29 16](#_Toc502824096)

[2017/12/30 16](#_Toc502824097)

[2017/12/31 16](#_Toc502824098)

[Week\_51 16](#_Toc502824099)

[2017/12/18 16](#_Toc502824100)

[2017/12/19 16](#_Toc502824101)

[2017/12/20 16](#_Toc502824102)

[2017/12/21 17](#_Toc502824103)

[2017/12/22 17](#_Toc502824104)

[2017/12/23 18](#_Toc502824105)

[2017/12/24 18](#_Toc502824106)

[Week\_50 18](#_Toc502824107)

[2017/12/11 18](#_Toc502824108)

[2017/12/12 18](#_Toc502824109)

[2017/12/13 18](#_Toc502824110)

[2017/12/14 19](#_Toc502824111)

[2017/12/15 19](#_Toc502824112)

[2017/12/16 19](#_Toc502824113)

[2017/12/17 19](#_Toc502824114)

[Week\_49 19](#_Toc502824115)

[2017/12/4 19](#_Toc502824116)

[2017/12/5 20](#_Toc502824117)

[2017/12/6 20](#_Toc502824118)

[2017/12/7 20](#_Toc502824119)

[2017/12/8 21](#_Toc502824120)

[2017/12/9 21](#_Toc502824121)

[2017/12/10 21](#_Toc502824122)

[Week\_48 21](#_Toc502824123)

[2017/11/27 22](#_Toc502824124)

[2017/11/28 22](#_Toc502824125)

[2017/11/29 22](#_Toc502824126)

[2017/11/30 23](#_Toc502824127)

[2017/12/1 23](#_Toc502824128)

[2017/12/2 24](#_Toc502824129)

[2017/12/3 24](#_Toc502824130)

[Week\_47 24](#_Toc502824131)

[2017/11/20 24](#_Toc502824132)

[2017/11/21 25](#_Toc502824133)

[2017/11/22 25](#_Toc502824134)

[2017/11/23 25](#_Toc502824135)

[2017/11/24 26](#_Toc502824136)

[2017/11/25 26](#_Toc502824137)

[2017/11/26 26](#_Toc502824138)

[Week\_46 26](#_Toc502824139)

[2017/11/13 27](#_Toc502824140)

[2017/11/14 27](#_Toc502824141)

[2017/11/15 27](#_Toc502824142)

[2017/11/16 28](#_Toc502824143)

[2017/11/17 28](#_Toc502824144)

[2017/11/18 29](#_Toc502824145)

[2017/11/19 29](#_Toc502824146)

[Week\_45 29](#_Toc502824147)

[2017/11/6 29](#_Toc502824148)

[2017/11/7 29](#_Toc502824149)

[2017/11/8 29](#_Toc502824150)

[2017/11/9 30](#_Toc502824151)

[2017/11/10 30](#_Toc502824152)

[2017/11/11 31](#_Toc502824153)

[2017/11/12 31](#_Toc502824154)

[Week\_44 31](#_Toc502824155)

[2017/10/30 31](#_Toc502824156)

[2017/10/31 31](#_Toc502824157)

[2017/11/1 32](#_Toc502824158)

[2017/11/2 32](#_Toc502824159)

[2017/11/3 32](#_Toc502824160)

[Week\_43 32](#_Toc502824161)

[2017/10/23 32](#_Toc502824162)

[2017/10/24 33](#_Toc502824163)

[2017/10/25 33](#_Toc502824164)

[2017/10/26 33](#_Toc502824165)

[2017/10/27 33](#_Toc502824166)

[2017/10/28 34](#_Toc502824167)

[2017/10/29 34](#_Toc502824168)

[2017/10/30 34](#_Toc502824169)

[Week\_42 34](#_Toc502824170)

[Plan 34](#_Toc502824171)

[Require 34](#_Toc502824172)

[Done 34](#_Toc502824173)

[2017/10/16 35](#_Toc502824174)

[2017/10/17 35](#_Toc502824175)

[2017/10/18 35](#_Toc502824176)

[2017/10/19 35](#_Toc502824177)

[2017/10/20 36](#_Toc502824178)

[2017/10/21 36](#_Toc502824179)

[2017/10/22 36](#_Toc502824180)

[Week\_41 36](#_Toc502824181)

[2017-10-09 37](#_Toc502824182)

[2017-10-10 38](#_Toc502824183)

[2017-10-11 38](#_Toc502824184)

[2017-10-12 38](#_Toc502824185)

[2017-10-13 38](#_Toc502824186)

[2017-10-14 39](#_Toc502824187)

[Week\_39 39](#_Toc502824188)

[2017-09-25 39](#_Toc502824189)

[2017-09-26 39](#_Toc502824190)

[2017-09-27 40](#_Toc502824191)

[2017-09-28 40](#_Toc502824192)

[2017-09-29 40](#_Toc502824193)

[2017-09-30 40](#_Toc502824194)

[Week\_38 41](#_Toc502824195)

[Plan 41](#_Toc502824196)

[Require 41](#_Toc502824197)

[2017-09-18 41](#_Toc502824198)

[2017-09-19 41](#_Toc502824199)

[2017-09-20 41](#_Toc502824200)

[2017-09-21 42](#_Toc502824201)

[2017-09-22 42](#_Toc502824202)

[2017-09-23 42](#_Toc502824203)

[2017-09-24 42](#_Toc502824204)

[Week\_37 43](#_Toc502824205)

[Plan 43](#_Toc502824206)

[Require 43](#_Toc502824207)

[2017-09-11 43](#_Toc502824208)

[2017-09-12 44](#_Toc502824209)

[2017-09-13 44](#_Toc502824210)

[2017-09-14 44](#_Toc502824211)

[2017-09-15 44](#_Toc502824212)

[2017-09-16 45](#_Toc502824213)

[2017-09-17 45](#_Toc502824214)

[Week\_36 46](#_Toc502824215)

[Plan 46](#_Toc502824216)

[Require 46](#_Toc502824217)

[2017-09-04 46](#_Toc502824218)

[2017-09-05 46](#_Toc502824219)

[2017-09-06 46](#_Toc502824220)

[2017-09-07 47](#_Toc502824221)

[2017-09-08 48](#_Toc502824222)

[2017-09-09 48](#_Toc502824223)

[2017-09-10 48](#_Toc502824224)

[Week\_37 49](#_Toc502824225)

[引用 51](#_Toc502824226)

[附录 52](#_Toc502824227)

[0000-年模板 53](#_Toc502824228)

[Week\_00-周模板 53](#_Toc502824229)

[Plan 53](#_Toc502824230)

[Require 53](#_Toc502824231)

[Done 53](#_Toc502824232)

[0000-01-01 53](#_Toc502824233)

[0000-01-02 53](#_Toc502824234)

[0000-01-03 53](#_Toc502824235)

[0000-01-04 53](#_Toc502824236)

[0000-01-05 53](#_Toc502824237)

[0000-01-06 54](#_Toc502824238)

[0000-01-07 54](#_Toc502824239)

# Overview

# 2018

## Week\_01

2017Q4绩效 2018-01-08

2018Q1绩效 2018-01-16

2017Y 绩效 2018-01-10

融合问题（已解决）

回复老葛邮件（已处理）

统计硬盘使用

### 2018/1/1

### 2018/1/2

1. 整理新疆数据采集工作记录并总结遇到的问题
2. 查询oracle的资源限制方法及rownum查询是否导致oracle服务shutdown
3. 参加辉哥关于新疆的工作会议
4. 梳理数据采集oracle采集语句相关的代码

### 2018/1/3

1. 配合系统第二轮测试
   1. 准备及放置资产层安装包
2. 新疆工作梳理
3. 向张思为交接新疆现场的工作环境及工作概况
4. 查找oracle性能相关文档（2 Day + Performance Tuning Guide）、了解oracle性能相关报告（awr、ash）及报告生成方式

### 2018/1/4

1. 配合系统测试
2. 修改数据采集代码，增加以下功能支持：
   1. 实现直接select（非rownum语法）采集功能
   2. 实现参数控制可选直接select采集功能运行
3. 指导张思为进行新疆数据采集工作
4. 与王辉沟通新疆数据采集相关方案及潜在问题

### 2018/1/5

1. 协调邢金彪、赵尚宇进行数据where etltime采集具体实施
2. 处理舟山hdfs数据导入数据仓库问题
3. 处理新疆现场需要查看采集语法sql事项
4. 测试简单采集语法（不带rownum语法）的功能逻辑是否正常达到，测试通过
5. 编译sparta项目，放在ftp上面；时间20180105

### 2018/1/6

1. 开发数据基于etltime密度分布功能
2. 向赵尚宇讲解新疆数据采集具体事项

### 2018/1/7

## Week\_02

### 2018/1/8

### 2018/1/9

1. 整理现场数据采集情况
   1. 目前2张表损坏，无法进行数据采集
      1. UT\_卡口\_人证合一\_检查站
      2. UT\_警综\_巡逻盘查人员
2. 元数据进行重新采集
   1. 新增表（或视图）390
   2. 进行新增表数据条数统计，总数据量（141亿）
      1. 1kw-5kw：39
      2. 5kw-1ww：13
      3. 1ww-10ww：15
      4. 10ww+
         1. 14ww：卡口\_环疆卡口经过车辆
         2. 19ww：卡口\_经过车辆\_源头
         3. 47ww：电力\_揍码数据表\_每半小时
3. 统计现场存储使用情况
   1. 机器磁盘：
      1. 001:
         1. 3.1T 90%
      2. 002
         1. 2.9T 84%
      3. 003
         1. 2.9T 83%
      4. 004
         1. 2.9T 82%
   2. Hdfs
      1. /Sparta 3.5 10.6
         1. /Sparta/moirai 85.6G 256.8G
         2. /Sparta/vulcanus 586.1G
            1. /Sparta/vulcanus/load 2286.1G
            2. /Sparta/vulcanus/core
            3. /Sparta/vulcanus/merge
      2. /hbase 166.7G 502.1G
      3. /user 50.9G 152.6G
4. 更新现场代码，并进行编译

### 2018/1/10

### 2018/1/11

### 2018/1/12

### 2018/1/13

### 2018/1/14

## Week\_03

1. **数据采集工作（保证目前采集的数据没有问题）**[与辉哥沟通过的]
   1. 全量采集（年后）
      1. **历史数据如何采集（出解决方案）**
      2. **删除重建表的采集（出解决方案）**
   2. 增量采集（年前）[已发操作email给思为]
2. 协助算法配置统一视图
   1. 现场配合算法配置（一周）
   2. 公司进行统一视图培训
3. 保证转换清洗与融合工作顺利进行[与辉哥沟通过的]
4. 集群扩容（待安排）
5. Etlcore处理结果数据（csv文件），column all null（某列数据全空） 判断处理 （本周）
6. 视图矩阵view.excel 处理报告 （1人两周）

### 2017/1/15

1. 整理出差工作事项\总结及规划数据资产当期工作
2. 处理系统测试建议、缺陷（和锋哥沟通后，缺陷统一解决）
   1. 组内沟通及系统测试沟通协调解决
3. 处理etltime 条数密度侦测现场异常
   1. ORA-01000: maximum open cursors exceeded
   2. 数据条数query时，其resultSet未close
4. 处理夏令时相关知识\处理新疆现场，“某兵团”数据融合未出现

夏令时、冬令时问题：

UTC与GMT时间表示

我们经常会看到UTC+8和GMT+8这样的时间显示。其实这都表示中国北京的时间，解释起来就是格林尼治平均时+8个小时，时区差东为正，西为负。我们中国处于东时区，也就是正数。

前面我们说的美国和俄罗斯两个国家的时区分别处于GMT-5和GMT+3时区的所以跟我们本地的时间有差别，比如莫斯科的GMT+3,中国的GMT+8，就是说，中国的时间比莫斯科的快5个小时（8-3）。但美国纽约所使用的GMT-5本来应该和中国GMT+8差13个小时，可为什么有的时候我们看到的却是差12个小时呢？这里就涉及到另一个问题了：[夏令时](http://www.trader007.net/tags/%E5%A4%8F%E4%BB%A4%E6%97%B6/)。

夏令时与冬令时

夏令时比标准时快一个小时。例如，在夏令时的实施期间，标准时间的上午10点就成了夏令时的上午11点。  


问题：从身份证号截取出生日期，比如：

1. 330902199005030014
   1. 出生日期：19900503
   2. 由字符串19900503得到时间，需要两个信息：
      1. 时区（从身份证中不可知该信息）
      2. 19900503

### 2017/1/16

1. 编译项目源代码
2. 向赵工介绍资产组这边主要数据处理流程以及讲明column全列是否all null的开发需求
3. 参加多维项目结项总结会议
4. 简单实现数据转换清洗报告，确定技术实现大致可行

### 2017/1/17

1. 书写增量采集操作文档并发email给思为
2. 协助新疆现场处理问题

### 2017/1/18

1. 协助现场进行部分问题分析与解决
2. 开发规则参数提取功能逻辑

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | Sqldeveloper | 采集的数据 |
| CZRQ | 操作日期 | Varchar2 | 有值（日期时间） | 全为null |
| DDCFRQ | 当地出发日期 | Varchar2 | 有值（日期时间） | 只有日期 |
| DDCFSJ | 当地出发时间 | Varchar2 | 有值（日期时间） | 只有时间 |
| DDDDSJ | 当地到达时间 | Varchar2 | 有值（日期时间） | 只有时间 |
| DDDDRQ | 当地到达日期 | Varchar2 | 有值（日期时间） | 只有日期 |

### 2017/1/19

1. 与赵工沟通数据资产这边的项目细节[整个流程已交代完毕并初步掌握]
2. 协助现场解决空间问题，结果：每台空余1.2-1.3T
3. 协助思为进行现场增量采集配置实现

|  |
| --- |
| 赵工，你好；  由于下周我去新疆出差；所以这两天（周五、周六），你那边最后先把项目（按邮件操作）先checkout下来，然后具体看看、先捋一下，有问题的话直接skype找我；理得差不多的话，我再在你那边给你代码简单走读下（当然看你那边的理解程度而定）；  我这边的使用技术大致为：  Scala、spark（2.0）  您看这样可行？ |
| 1. 大数据开发环境 2. 数据处理流程介绍（涉及表结构、代码执行入口） 3. 具体流程案例    1. 数据采集、数据转换清洗、融合    2. 转换清洗数据导入到spark-sql中 |
|  |

### 2017/1/20

Git 异常解决

|  |
| --- |
| 1. this exceeds GitHub’s file size limit of 100 MB    1. git filter-branch --force --index-filter "git rm --cached --ignore-unmatch filePath" --prune-empty --tag-name-filter cat -- --all    2. git commit --amend -CHEAD    3. git push origin master |
| 1. Cannot rewrite branches: You have unstaged changes.    1. git status    2. git add --all |
| 1. git commit –m “message” |
| 1. git push original master |

### 2017/1/21

## Week\_04

### 2017-01-22 -> 2017-01-28

|  |
| --- |
| 总结，按计划完成：  1. 增量采集定时任务部署及测试  2. 转换清洗及融合定时任务部署及测试  3. 排查处理遗留问题“维族轨迹—民航订票”；具体见下述“3. 待处理问题”  4. 现场搭建部署spark-sql可视化操作平台  5. 与葛挺组联调增量数据处理流程  以下是此次出差事项总结：   1. 协商统一视图更新，(2018-1-23 11:42:44)    1. 原因：上周由于统一视图多值选项算法配置出错，算法想确定融合多值是否满足需求    2. 结果：由于更新统一视图，会导致后续数据处理重跑；经与王辉、葛挺沟通，**暂不做处理@王辉**，在**家里重点测试融合多值实现@张思为**    3. 引申：葛挺表示**多值分割符有问题**，我们这边应该测试多值的值解析没有任何问题；@张思为    4. 可以进行相关结果截图，表明此功能没有问题@张思为    5. 家里已测试通过；确定多值没有问题 2. **处理增量采集表**    1. 查询增量表表ID及etltime属性ID【23张】    2. 编写workflow    3. 测试workflow       1. 问题列表：          1. Job.properties被改错了如下项：             1. Namenode，错配为路径名             2. Spark\_home，错配为单机版    4. 采集workflow提交为定时任务 3. 待处理问题    1. UT\_部科信\_维族轨迹\_民航订票       1. CZRQ全为null（源表sqldeveloper非null）       2. 日期、时间采集过来表达为实际，而sqldeveloper看起来为日期时间    2. 问题解决       1. 与三汇沟通，问题还是存在、一致（与sqldeveloper一致）       2. 书写简单jdbc采集代码进行头10条数据验证（问题存在、一致）       3. 将该表数据导入spark-sql中；发现数据在某段时间内是czrq为空、ddcfrq是日期、ddcfsj是时间；在另外时间段内czrq均是日期时间、ddcfsj与ddcfrq也都是日期时间    3. **总结：该问题实际不存在，该问题是因为使用数据时对数据未做全面了解，验证问题是否存在时，也未作全面验证导致**；**实际采集的数据与Oracle数据完全一致；**只是数据本身存在格式不一致以及czrq null的情况（czrq 363万为null，数据2kw） 4. 运行转换清洗、运行融合 5. **转换清洗融合任务workflow、coordinate、properties编写；测试及提交为定时任务** 6. 在现场搭建spark-sql可视化操作平台（具备notebook类似的功能，可执行sql语句并得到结果以图表形式展现）[这块会在进行技术分享]    1. 验证数据处理性能，在5kw（某一个月的住宿记录）groupby 酒店名称，2-3分钟就得出了结果    2. 地址：duowei001:8686 7. **与葛挺组联调增量数据处理流程** |

## Week\_05

2018-01-29 -> 2018-02-04

|  |
| --- |
|  |

|  |
| --- |
| 1. xj现场，源数据某列有值，转换清洗未加规则而融合全部为null的问题  a. 怀疑为新产生的数据（贴源层），转换清洗未执行  b. 这些数据（目标）表对应多表输入，其中该列有值（贴源层）的行其数据不合格 |
| 提出部分融合性能优化办法：  1. 拍照xj融合运行链路日志  2. 尽量分析链路日志运行时间  3. 融合事件直接进行文件转移，不进行数据全量读入 |
| 融合job oozie运行资源重新配置  1.  2. 根据脚本来改变oozie coreMerge任务的配置  spark\_opts=--num-executors 3 --executor-cores 1 --executor-memory 1G --driver-memory 512m --conf spark.yarn.historyServer.address=${historyServer} --conf spark.eventLog.dir=${nameNode}/var/log/spark\_hislog --conf spark.eventLog.enabled=true  3. ozoie kill 任务：oozie job -oozie  [http://localhost:11000/oozie -kill 14-20090525161321-oozie-joe](http://localhost:11000/oozie -kill 14-20090525161321-oozie-joe4)  [4](http://localhost:11000/oozie -kill 14-20090525161321-oozie-joe4). oozie运行任务：oozie job -oozie http://localhost:11000/oozie -config job.properties -run |
| 转换清洗某列全为null报告：  Case class TableReport(originTableId, targetTableId, isRight, path, count, Array[AttributeReport]())  Case class AttributeReport(attrId, attrName, notNullCount, isAllNull)  结果路劲：参数前缀+目标表名+源表表名+文件名（tableMapping= tableMappingid） |
| 算法张凌关于最终使用统一视图版本的问题 |
| 力行者 激励在一线支撑或维护，并取得良好业绩或良好客户反馈的同事。    小蜜蜂奖 激励在数据采集、数据工具/平台、清洗方案，数据质量、数据服务取得良好业绩的团队。 |
| 1. 面对新疆三汇约束采集方案突然变更（数据量级：百亿），在3天内进行快速响应开发及测试，提供了另一套自定义区间采集的功能，并且进一步实现了自动侦查数据密度采集功能；顺利完成了现场数据采集；3天内快速验证及部署增量采集、转换清洗、融合任务  2. 在一周内开发完成数据转换清洗报告、规则验证，提供列粒度数据经过（清洗转换）规则的具体细节报告，比如格式化dateStr，经过规则dateStr2LongRule，得到输出时间戳；解决原有人工一一验证（查看文本分割符结果数据）耗时长、重复、繁琐等问题  1. Spark-sql数据仓库，存储格式由分隔符文本变为parquet；数据类型进行了统一、存储空间降低了3-20倍、提供了sql数据处理（探索分析）接口（SQL）命令行及web notebook界面）  2. 数据质量模块，提供列粒度的高频值、缺失率、数学类型的最大、最小、方差、平均数等指标的统计及侦测客户数据条数变更并形成折线图（条数、日期）  3. 在一周内开发完成数据转换清洗报告、规则验证，提供列粒度数据经过（清洗转换）规则的具体细节报告，比如格式化dateStr，经过规则dateStr2LongRule，得到输出时间戳；解决原有人工一一验证（查看文本分割符结果数据）耗时长、重复、繁琐等问题 |
| 1. 改良数据质量数据条数变更查询功能，增加查询超时时间 |
|  |
| 1. 和思为商议资产新疆接下来相关主要工作（确定算法统一视图对接人员），介绍统一视图的3个辅助工具  2. 了解spark dataframe takeSample功能细节，确定能否用于转换清洗报告  3. 与赵尚宇工再次确认列全为null结果输出具体形式  4. 探究scala文件结果输出的几种方式，确定一种简单实用的方式 |
| 1. 完善、修改列全为null功能实现  a. 增加dataframe处理封装层  b. 增加结果转换为json并输出到文件  c. 修改报告结果部分封装case class |
|  |

## Week\_06

|  |
| --- |
| 数据采集代码逻辑、代码整理 |
| 平台技术规划、设计 |
| 辅助功能整理开发 |

## Week\_07

## Week\_08

## Week\_09

## Week\_10

## Week\_11

## Week\_12

## Week\_13

## Week\_14

## Week\_15

## Week\_16

## Week\_17

## Week\_18

## Week\_19

## Week\_20

## Week\_21

## Week\_22

## Week\_23

## Week\_24

## Week\_25

## Week\_26

## Week\_27

## Week\_28

## Week\_29

## Week\_30

## Week\_31

## Week\_32

## Week\_33

## Week\_34

## Week\_35

## Week\_36

## Week\_37

## Week\_38

## Week\_39

## Week\_40

## Week\_41

## Week\_42

## Week\_43

## Week\_44

## Week\_45

## Week\_46

## Week\_47

## Week\_48

## Week\_49

## Week\_50

## Week\_51

## Week\_52

# 2017

## Week\_52

### 2017/12/25

### 2017/12/26

1. 整理新疆现场数据采集概况，同算法同事交流其最新的数据采集需求；
   1. 总共分4批，具体为属性表、行为表
2. 分表采集数据及验证数据采集是否正确
   1. 采集（带rownum）正确表为86张，其中亿级别表6张
   2. 未成功采集表8张，数据量级分别为，1-3亿：5张；4亿：1张；10亿：1张
   3. 由于三汇明确要求不准用rownum语法采集（rownum语法是并行采集的关键，也是oracle数据采集的核心技术；当初选定该技术，是因为本地性能测试对比不用该语法速度快10倍左右）；故13:48停掉了所有的采集任务
3. 整理现场环境及文档，方便后续人员尽快进入工作
   1. 转移邱工出差的记录文件到我们的机器上
   2. 整理我这次出差的工作记录文档及相关文件

### 2017/12/27

### 2017/12/28

### 2017/12/29

### 2017/12/30

### 2017/12/31

## Week\_51

1. 数学oozie任务配置提交文档，书写一个get started，3张增量表示例
2. 配合展丽霞支持算法方面关于hbase相关（老葛提过2次，2017-12-19 10:05:03）

### 2017/12/18

1. 解答系统测试人员相关问题
   1. 关于spark-sql部署（mysql元数据库）
   2. 关于任务重跑，删了mysql中的处理记录，hdfs相应数据未删除，导致运行不成功
   3. 关于增量采集细节，配置增量字段后（a. 改表已采集；b. 改表未采集）
2. 解决ide右键测试方法名没有run的问题，原因junit与scalatest在同一class混用
3. 验证新的发布版本（解决集成测试缺陷）
   1. 测试部署、数据库连接信息配置、元数据采集、数据采集、转换清洗、融合、hdfs text到spark-sql、数据质量 均通过
   2. 0.7集成测试建议也已修复

### 2017/12/19

1. 结合邱工的现场信息处理新疆数据采集的相关问题
2. 参加项目周会
3. 在80集群上面部署及运行资产组测试项目；用于验证0.7缺陷解决完毕
   1. 融合运行脚本在单机正常，集群状态下执行不成功，已找到问题根源
4. 进行代码review

### 2017/12/20

1. 结合邱工的现场信息处理新疆数据导出异常，number类型precision与scale过长
   1. 确定该缺陷为spark官方缺陷SPARK-20427，得到2.3.0版本解决，目前未发布
   2. 在本地重现现场的异常场景
   3. 寻找该缺陷的解决方案，在oracle中创建view；query语句实现cast转换
   4. 验证query cast语句方案是否可行，确定该方案可行
   5. 修改采集代码，将验证的方案在采集中实现，并测试通过
   6. 配合现场采集模块进行升级
2. 处理系统测试用例缺陷
3. 解答算法关于hbase的resultScaner、result及cell的相关问题

### 2017/12/21

1. 在服务器上解决及验证spark任务主类传递参数带空格技术问题
   1. 脚本运行方式，传递参数及”$1”均需用””进行封装
   2. Spark-submit运行方式，传递参数均需用””进行封装
2. 参加统一视图会议
3. 协助邱工解决现场问题
   1. 分批采集where语句如何使用
4. 协调系统测试处理问题
   1. DFK\_GAXZ元数据采集表象失败
   2. 排查系统测试资产这边目录被更改

### 2017/12/22

1. 给予系统测试svn版本信息
2. 支援邱工现场，本地分析现场oracle服务关闭
   1. 查找oracle内存监控方案
   2. 采集大批量数据，监控oracle的内存变动；并发现其规律
   3. 发现oracle的session对应采集core的内存最大为1.7M，内存变动不大
   4. 协助愈工杀死现场环境的数据质量条数定时统计任务
   5. 思考并解答邱工关于分批采集的具体实施细节
3. 处理军伟的数据需求，插入4条打防控人与案件关联的记录
   1. 分析具体在那张源表里面插入数据记录
   2. 插入数据记录，重跑跑采集、转换清洗、融合

|  |
| --- |
| 422822198603054031:A3309226900002004080004  330903198806291814:A3309216100002004040006  510213198203291215:A3309216100002004040006  330881198912261524:A3309216100002004040006 |

### 2017/12/23

### 2017/12/24

## Week\_50

1. 跟踪新疆现场数据采集情况，数据写入到spark-sql数据仓库

### 2017/12/11

1. 整理新疆出差数据采集总结文档（已通过邮件发出）
2. 修改现场遇到到问题：数据条数区间统计功能支持
3. 测试汉字转拼音技术可行性，确定汉字转拼音技术可行
   1. 实现中文汉字转拼音功能，测试该功能通过
4. 测试及增加数据采集到spark-sql，中文表名转拼音技术支持
   1. Oracle中新建中文汉字表名的表并插入数据，Mysql中插入相应的元数据记录
   2. 修改采集源代码，实现中文表名转拼音
   3. 运行及测试中文表名全流程，测试通过
5. 与集成测试人员沟通80上的数据资产这边的部署情况

### 2017/12/12

1. 数据质量模块，中文表名转拼音技术支持
   1. 运行及测试中文表名全流程，测试通过
2. 解答系统测试关于转换清洗结果怎样找相应源表数据的问题
   1. 如何验证转换清洗是否正确
   2. 关于清洗如何实现、具体清洗规则的含义
3. 关于vulcanus全流程支持中文实现细节的讨论
4. 数据从hdfs到spark-sql模块中文表名技术支持
5. 部署数据资产组项目，配置及采集元数据
6. 新疆现场数据采集任务重新挂起执行（由于现场机器网络问题，程序终端）

### 2017/12/13

1. 实现及测试大数据公共模块：hdfs文本数据迁移到spark-sql，表名中文功能技术支持
2. 实现数据采集count不统计及count自定义参数传入功能
3. 确定新疆现场数据采集情况，还有14张表数据待采集

### 2017/12/14

1. 解决锋哥运行数据质量任务所报的异常
   1. 一个为hdfs数据类型过于繁杂，数据质量需要逆解析；类型不支持
   2. 一个为spark-sql原生sql过长语句解析异常，不影响功能
2. 系统测试用例评审
   1. 沟通hdfs数据源采集数据写到spark-sql资产池
   2. 沟通hdfs数据源采集数据数据质量统计问题
3. oracle数据采集实现自定义where条件参数外传
4. 确定转换清洗基于0.7开发写入spark-sql模块中文表名技术支持
5. 与系统测试及集成测试沟通数据质量抽样阈值相关

### 2017/12/15

1. 解决集成测试0.7版本数据质量数学类型结果表结果多余列建议缺陷
   1. 修改db模块相关代码及xml
   2. 修改数据质量相关代码
   3. 测试在既有表结构上运行成功，测试在表结构修正上运行成功
2. 解答系统测试关于资产组运行日志相关问题
3. 解答集成测试关于merge数据质量数学类型没有结果问题
4. 解决集成测试转换清洗运行异常问题，排查为脚本类名未改变

### 2017/12/16

### 2017/12/17

## Week\_49

### 2017/12/4

1. 实现锋哥提出源头添加固定身份证、车牌数据的需求
   1. Oracle插入携带身份证、车牌的数据
   2. 删除采集、转换清洗、融合记录及hdfs数据
   3. 重新执行采集、转换清洗、融合
   4. 配合前端一起验证该需求实现
   5. 重新再oracle中插入携带车牌信息的数据（由于验证发现车辆ws接口不可修改）
   6. 书写对应相关文档
2. 测试0.7版本的采集、转换清洗、数据质量[基于spark-sql dataframe开发]
   1. 修改相关脚本文件及jar数据库、spark-sql配置
   2. 采集测试成功
   3. 数据质量测试成功
   4. 转换清洗测试成功
3. 修改vulcanus用户手册
   1. 添加数据质量部分功能
   2. 修改hdfs text数据到spark-sql数据仓库部分
4. 修改源代码（采集、webservice采集、bigdata、数据质量、转换清洗、融合）
   1. 修改源代码hive-site.xml模块；仿照mysql配置，实际部署时不需要修改

### 2017/12/5

1. 帮助张思为解决融合0.7任务运行失败问题
   1. 因为脚本--driver-java-options该选项导致（删除该选项解决）
2. 书写oozie使用文档，主要介绍
   1. workflow.xml\ coordinator.xml\ job.properties\任务job提交\增量采集workflow生成
3. 实现oracle表16数据（已采集）重新采集（金龙提出的）
4. 集成测试数据资产这边基本环境信息（安装目录、mysql db）介绍
5. 协助思为部署系统测试环境（代码、spark-sql）
6. 在笔记本上重新构建开发环境
   1. 插上网线编译项目
   2. 拔掉网线编译项目

### 2017/12/6

1. 出差

### 2017/12/7

1. 询问及整理现场环境
2. 准备及传输数据（程序及文档），通过申请再进行拷贝
3. 验证现场oracle环境，与算法人员沟通得出实际的连接信息
4. 部署vulcanus项目（系统环境配置，数据库初始化、元数据采集、数据采集、数据条数探测等）
5. 进行配置之后的元数据（表及表结构信息）采集，并与算法人员沟通是否正确
   1. 240张源表
6. 执行数据采集

### 2017/12/8

1. 执行数据质量条数统计
   1. 目前有2张表条数统计未出结果
   2. 其余表全部统计出结果，总条数约为120亿
   3. 调整先执行表条数小的数据采集（表条数多的执行时间过长，放在后面采集）
   4. 目前已采集的数据表数为：211，表条数总计约为：21亿
   5. 待采集（包括正在采集的）表为：20张（其他9张条数为0或者表不存在）
2. 将未采集的表信息发给算法人员，与算法人员确定未采集的表中那些表需优先采集
   1. 提交需要优先采集表的采集任务

### 2017/12/9

出差

### 2017/12/10

## Week\_48

Oozie sharelib问题；运行sparksession没有问题（自己编译sharelib），但是 enableHiveSupport()后；提交到sparksql失败

1. 解决NoClassDefFoundError: org/apache/hadoop/hive/conf/HiveConf$ConfVars
2. Vulcanus可用版本发布到ftp服务器
3. 部署文档
   1. Solr部署文档
   2. Sparksql数据仓库部署文档
   3. Oozie部署文档
   4. Vulcanus项目部署文档
4. 功能自测
   1. 采集功能
      1. 全量采集
      2. 增量采集
      3. Hdfs文件采集
      4. 保存到hdfs
      5. 保存到spark-sql
   2. 数据质量
      1. 字段统计
      2. 条数定时探查
   3. 数据仓库
      1. 采集数据仓库
      2. 融合数据仓库
   4. 任务调度
      1. 增量采集定时采集

### 2017/11/27

1. 排查Oozie sharelib问题；运行sparksession没有问题（自己编译sharelib），但是 enableHiveSupport()后；提交到sparksql失败
   1. 解决当前异常：java.lang.ClassNotFoundException: org.apache.hadoop.hive.conf.HiveConf$ConfVars
   2. 遇到新异常：org.datanucleus.exceptions.NucleusUserException: The connection pool plugin of type "BONECP" was not found in the CLASSPATH!
      1. 目前处理了半天，没有获得任何头绪

11/27

1. 处理出差期间积压的邮件.

2. 专利检索查看反馈.

3. 专利交底书修改提交.

4. vulcanus数据融合0.6遗留bug定位.

5. vulcanus任务变化讨论会.

### 2017/11/28

1. 参加0.7版本测试评审
2. 解决现场hdfs文本数据到数据仓库
   1. 准备实际操作文档及邮件
   2. 准备实际可用的jar包及运行脚本
3. 排查Oozie sharelib问题；新异常：org.datanucleus.exceptions.NucleusUserException: The connection pool plugin of type "BONECP" was not found in the CLASSPATH!
   1. 验证解决方案，已解决
4. 参加系统测试会议

### 2017/11/29

1. 回答系统测试关于数据资产这边的相关问题
   1. 数据资产这边的主要功能及处理流程
   2. 各功能的基本介绍
2. 我们这边今天提供3个文档[已提供]
   1. 统一视图里代码实现的规则说明文档
   2. 书写增量字段配置
   3. 书写数据质量结果说明
   4. 把0.5概要设计提供采集、转换清洗、融合提供给系统测试人员来说明资产这边的主干功能
3. 按照测试讲的来自测，达成共识

### 2017/11/30

1. 新疆数据采集
   1. 得有spark 2.x运行环境
   2. 元数据采集模块必须提前运行并得到结果
   3. 源数据可以被当前vulcanus采集程序进行采集（得事先确定）
      1. 目前本地仅支持oracle数据源、hdfs（元数据得能够采集，符合一定规则）
   4. 目前这边支持数据保存到hdfs的文本文件及spark-sql数据仓库
   5. 增量采集必须事先配置增量采集字段（否则默认全量采集）
   6. 如果确定以上所述具备，按以下步骤进行数据采集：
      1. 采集元数据步骤：
         1. 配置数据源信息：
         2. 配置数据源实例连接信息：
         3. 进行元数据采集
      2. 配置增量字段[可暂时忽略]
      3. 运行采集数据脚本
2. 书写新疆数据采集相关文档
3. 协助展丽霞调整现场solr内存
4. 测试数据质量模块
   1. 解决数学类型字段长度为0的异常，数学类型统一为类型默认长度
   2. 解决column根据实际unique\_count(之前为count)的大小与阈值比较进行分类异常
5. 测试数据采集hdfs模块
   1. 修复待采集text文本数据分隔符元数据配置缺失问题
   2. 验证采集模块oracle、hdfs、增量、保存到hdfs、保存到spark-sql等功能是否正常
   3. 修复hdfs文本数据采集保存到spark-SQL功能支持
6. 准备vulcanus整体测试
   1. 源表数据采集失败，按照金龙建议操作:重新签db及webservice模块代码重新编译打包
   2. 采集失败，由于数据源case加密导致的；重新编译打包解决

### 2017/12/1

1. 编译vulcanus可运行jar，确定发布版本的tar包（已更新到ftp）；

a. 配合龙伟进行部署测试

a. 测试数据质量模块（测试通过）

2. 配置集成测试环境

a. 配置spark-sql数据仓库环境，测试通过

b. 配置oozie任务调度环境，测试通过

3. 书写系统测试部署文档

a. vulcanus代码部署文档[测试通过]

b. spark-sql部署文档

c. oozie部署文档

d. solr部署文档

4. 回答集成测试同事关于vulcanus的测试问题

12/1

1. 四台系统测试服务器集群部署, 并进行功能和健壮性测试.

2. sparta集群安装部署文档增加更多的细节以便他人进行安装部署.

3. 和组件组沟通基于2.3.\*的集群扩容问题, 进行集群扩容尝试.(没来得及尝试)

4. 沟通解决数据融合相关问题.

5. 优化集成测试集群参数.

### 2017/12/2

### 2017/12/3

## Week\_47

问题：

1. 车辆，人脸任务运行时，spark（enableHiveContext）初始化不成功

2. spark（enableHiveContext）初始化不成功的任务在oozie上面运行不了，而spark（非dataframe保存sparksql版本）顺利运行

1. 现场增量字段配置 [他人]
2. 现场Oozie搭建 [自己]
3. 现场Oozie采集、转换清洗、融合任务部署及测试 [自己优先]
4. 现场车辆、人脸数据接入测试 [他人]
5. 现场数据重新采集进入spark-sql [他人]
6. 数据仓库搭建[自己]
7. Spark培训[已进行]

### 2017/11/20

1. 准备明天spark培训资料及设计具体实施
2. 任务调度模块添加生成workflow保存路劲前缀路劲参数传入功能
3. 初步确立交通过车记录数据schema
4. 代码实现交通过车记录信息查询

### 2017/11/21

1. 在笔记本上准备开发环境
   1. 安装intellij community,java,scala
   2. Checkout 源代码并进行编译打包,确保在现场也能够进行开发
2. 准备出差数据文件
   1. 增量采集字段配置相关资料
   2. Oozie部署相关,oozie workflow部署相关(增量采集全流程通过oozie进行定时执行)
   3. Spark-sql数据仓库相关;(现场环境已变更,需重新部署)
   4. 车辆、人脸数据接入相关资料
   5. 转换清洗更新相关资料
   6. 0.7版本数据存到spark-sql数据仓库
3. 实施spark实践简述及会后回答相关疑问

zhangsiwei6(张思为6) 11-21 20:08:55

1. 修改cm环境清理脚本。

2. 修改三台服务器网卡配置。

3. 安装cm并测试重新安装不会清理掉挂载磁盘上的数据。

4. 清理三台服务器集群环境。

5. 三台服务器初始化相关设置。

6. 发现larkServer问题， 跟组件组沟通解决。

7. 发现namenode起不来， hdfs格式化不成功问题， 正在跟组件组沟通解决。

8. 集群部署（进行中）。

### 2017/11/22

1. 准备此次出差需要的数据及文件到208上面

2. 解决现场集群部署不成功问题

1. 现场13机器的cm启动不起来，最终排查为java环境问题；重新安装java得以解决

2. 测试hbase表读写正常，从而推测集群暂时正常

3. 现场postman测试人脸、车辆webservice的数据采集接口

### 2017/11/23

1. 修复线上solr服务，案件地图模块

1. 排查solr问题

2. 修改配置：vim /etc/default/solr.in.sh ，改变zookeeper连接

3. 准备及上传solr配置相关文件到zookeeper环境

4. web页面测试solr成功修复

2. 重新部署spark-sql数据仓库

1. 准备mysql 源数据库hivedb环境

2. 改变相关配置并添加相关jar包

3. 测试创建数据库，mysql hivedb中增加相关记录，测试通过

3. 更新转换清洗相关

1. 更新hosts文件，加入mysql主机名映射

2. 启动webservice，更新统一视图

3. 执行转换清洗任务

4. oozie部署

1. 排查oozie配置数据库相关选项不生效的问题

2. 解决oozie配置使用mysql数据库问题

3. 安装部署sharelib

4. 测试wordcount workflow运行

### 2017/11/24

1. 车辆、人脸、webservice代码测试，数据取出正常；保存到sparksql异常

2. 增量采集任务

1. 配置增量采集字段，通过筛选增量表的字段判断增量字段为：'LRSJ'（录入时间）, 'RKSJ'（入库时间）, 'ZHGXSJ'（最后更新时间）, 'CSSJ'（数据传送时间）, 'SELLTIME'（售票时间）

2. 编排增量采集任务的oozie workflow，并提交运行

3. 执行0.7版本数据采集任务，将数据采集到数据仓库sparksql

4. 协助解决服务器磁盘重新挂载问题

### 2017/11/25

1. 整理出差文档
2. 修复车辆、人脸数据采集静态变量初始化导致值未改变的问题；
3. 修复车辆、人脸数据采集的数据保存到sparksql，sparksession初始化异常问题
   1. 排查为任务提交方式不对；原为java –cp；实际应该为spark-submit

### 2017/11/26

## Week\_46

1. 车辆、人脸数据接入
2. Spark技术分享
   1. Spark实践
      1. Python开发
   2. Spark-sql实践操作
   3. Scala，函数封装
   4. R可视化等
3. 数据质量熟悉

### 2017/11/13

1. 规划设计人脸、车辆数据接入具体代码实现方案
   1. 实现人脸数据接入流程图
   2. 实现车辆数据接入流程图
2. 数据质量条数变更数据结合源表目标表mapping数据进行分析处理
   1. 数据一个月条数变动5次以上，并且被目标表使用的源表有：1, 2, 3, 4, 9, 15, 19, 21, 27, 28, 35, 41, 50, 52, 86；总共15张表；具体图表见邮件

张思为6

1. 统一视图细节了解讨论.

2. 统一视图修改方案思考.

3. 熟悉已有的统一视图中的目标表结构.

4. 跟进案件目标表码表缺失问题.

5. 出差准备工作.

### 2017/11/14

1. 人脸抓拍库数据抓取由同步接口改为异步接口（3步）
   1. 原因：舟山现场同步接口不可用
   2. 测试异步接口流程可用性
2. 测试及确定webservice数据采集技术选型：okhttp
3. 参加spark on yarn技术分享

张思为

1. spark,yarn分享会.

2. 规则原理学习, 相关代码理解.

3. 数据迁移相关案例查看, 测试.

### 2017/11/15

1. 确定人脸抓拍库返回结果的schema，用于注册dataframe，及形成spark-sql表结构
2. 实现抓拍库数据账号重新赋值为给定查询账号，用于标识抓拍数据实际对应某人
3. 时间处理逻辑实现，格式化，往前n天，现场接口时间跨度小于30天；初始化最早开始采集时间等
4. 参加数据质量的会议

11/15

1. vulcanus部署脚本理解修改.

2. vulcanus部署过程中细节了解.

3. 创建并初始化表与统一视图对应了解.

4. webservice模块了解.

5. 数据质量问题讨论.

### 2017/11/16

1. 代码实现webservice get、post
2. 代码实现人脸静态库接口查询及主要参数封装（通过cardid获取人脸模型及图片url）
3. 代码实现人脸静态库以图搜图异步接口查询及主要参数封装
   1. 代码实现提交任务得到任务taskid 接口查询
   2. 进度接口查询、轮询
   3. 最终结果获取
4. 以上接口功能测试通过

11/16

1. 检查修改init等脚本写死的部分, 并测试.

2. 80服务器更新mysql,修改创建并初始化表sql文件,添加数据质量相关表并测试.

3. webservice模块了解.

### 2017/11/17

1. 实现webservice查询的结果数据输出到spark-sql
   1. 实现Json str buffer文件写入器，用于格式化将webservice json结果落地到spark-sql中
   2. 实现json str buffer写入器缓冲buffer池，避免频繁提交
2. 实现webservice数据采集mysql元数据操作封装
   1. 实现人脸、车辆对应的数据源、数据源case、源表数据设计构建
   2. 实现采集版本信息构造，startkey，endkey为采集对应的时间区间

张思为

1. 出差准备，集群部署测试，生成文档
2. 修改部分脚本测试部署

### 2017/11/18

### 2017/11/19

## Week\_45

1. 根据一批身份证号码获取一批证件照[王辉11]
2. 根据一批证件照从人脸webservice接口获取抓拍记录[未验证以图搜，已验证按时间区间搜抓拍记录]
3. 根据一批车牌号从车辆webservice接口获取过车记录[已验证车牌+时间区间搜过车记录]

|  |  |
| --- | --- |
| 采集 | 集群 |
| 车辆、人脸 | 融合 |
| 数据仓库 | 转换清洗【规则+统一视图】 |
| 数据质量、任务调度 |  |
| 项目打包部署 |  |

### 2017/11/6

1. 参加项目周会
2. 参加v0.7需求评审
3. 查找最接近的专利，并提交专利检索申请

### 2017/11/7

1. 解决文档缺陷：
   1. （"可以接入所有数据",过于绝对,建议更改）
   2. （"数据访问"部分,应提供scala api内部接口文档说明）
2. 简单理解转换清洗及规则应用模块

zhangsiwei6(张思为6)

1. 专利流程提交, 专利细节修改.

2. 学习理解转换清洗模块.

### 2017/11/8

1. 书写现场导mysql数据文档
2. 实现通过表id统计该表各列数据是否有序的功能逻辑
   1. 初步识别是否有序，字段有序特性将作为增量字段配置参照

张思为6

1. 集群参数调优设置.

2. 在集群部署vulcanus.

3. 之前版本bug解决.

1. 锋哥帮忙把现场机器的mysql数据库vulcanus导出来（也可以只导其中的表：**data\_load\_pioneer**）
   1. Mysql登录信息：
      1. Ip：10.123.128.208
      2. User：root
      3. Passwd：Hik12345+
2. 技术参考：备份mysql数据库或者某张表
   1. mysqldump -uroot -pPASSWD vulcanus > vulcanus.sql
   2. mysqldump -uroot -pPASSWD vulcanus data\_source\_case> data\_source\_case.sql
   3. mysqldump -uroot -pPASSWD vulcanus origin\_table> origin\_table.sql

### 2017/11/9

1. 处理数据质量已条数变更数据，用于增量采集配置
   1. 从现场拿到具体统计数据
   2. 条数在过去一个月的变动超过5次的表有115张
   3. 学习R的折线图可视化（因为条数变动有变大、变小；用折线图（x=天日期，y=条数）可以很直观的了解数据增长趋势）

张思为6

1. 数据融合bug解决.

2. 转换清洗模块学习.

### 2017/11/10

1. 处理数据质量已条数变更数据，用于增量采集配置
   1. 将数据进行折线图（x=天日期，y=条数）可以很直观的了解数据增长趋势）图形化，具体图表见邮件
2. 数据质量字段有序统计【用于辅助增量采集字段配置】
   1. 实现通过传入tableid文件进行统计、有序字段结果文件输出
   2. 本地测试数据质量字段有序统计功能，测试通过

张思为6

1. 数据资产池部署熟悉。

2. 统一视图了解及原理学习。

### 2017/11/11

### 2017/11/12

## Week\_44

### 2017/10/30

1. 进一步修改及测试hdfs模块的parquet及avro基于spark-sql开发的功能
2. 书写增量任务调度、增量数据采集俩更新模块的概要设计文档

10/30

1. 数据融合基于dataframe开发.

2. 新版本数据融合测试.

数据融合优化 80%

融合基于spark-sql dataframe开发 100%

### 2017/10/31

1. 实际验证测试人脸webservice以下接口：
   1. 通过证件号码获取证件照（证件照模型）：静态库人员信息和模型查询
   2. 通过照片进行建模：提交人脸建模
   3. 根据人脸模型到抓拍库搜索抓拍记录：抓拍库以图搜图同步接口

zhangsiwei6(张思为6) 10-31

10/31

1. 数据融合本地测试.

2. 数据融合服务器测试.

3. 数据融合数据查看.

4. 代码review.

数据融合优化 100%

融合基于spark-sql dataframe开发 100%

likai14(李凯14) 10-31 19:14:32

融合 的速度较慢，与0.6版本一样（加了优先级）

likai14(李凯14) 10-31 19:14:47

思为 计划在0.8版本解决

### 2017/11/1

1. Spark json数据处理，尝试解决json array 自动转换为dataframe；
   1. 因为发现webservice返回的数据均为json格式；所以如果json 2 dataframe技术通过，数据处理链路将会变得简单高效

### 2017/11/2

1. 解决同事定制化spark-sql命令行控制台自定义日志输出级别；
   1. 去除干扰日志，使数据输出更清楚简洁
2. 解决json array形成的dataframe，由于json对象属性不对齐导致的异常
   1. Json webservice返回结果，有些值为空直接不放回；故json schema不完整

### 2017/11/3

1. 组内代码review相关
2. Review自己开发模块代码：采集代码、数据质量，简化逻辑，添加异常处理，添加注释等

## Week\_43

数据仓库使用方联调问题

多多询问思为工作中遇到的技术问题

1. 确认，即使数据已限定为舟山市；百度转换的结果经纬度还是不准，并且偏移出舟山。

### 2017/10/23

数据仓库概要设计文档 100%

1. 完成数据仓库概要设计的模块、流程图
2. 完成数据仓库概要设计书写

### 2017/10/24

1. 重构后的增量采集功能测试 50%
2. 向张思为讲解链路日志的路劲形成规则，即如何解析
   1. 介绍spark-sql数据如何读取（主要讲解etl-core对应的数据读取及保存部分代码）

### 2017/10/25

1. 重构后的增量采集功能测试 100%
   1. 实现增量采集与oozie结合实现定时采集，并测试通过

张思为6

10/25

1. 数据融合基于dataframe开发.

融合基于spark-sql dataframe开发 60%

### 2017/10/26

1. 解决oracle Sparta连不上问题
2. 解决转换清洗输出表，表名没带源表表名问题

增量采集的数据与转换清洗、融合整个处理链路进行集成及测试 30%

zhangsiwei6(张思为6) 10-26 18:24:02

10/26

1. 发现转换清洗输出路径问题, 讨论解决方案.

2. 数据融合基于dataframe开发.

数据融合优化 50%

融合基于spark-sql dataframe开发 70%

### 2017/10/27

1. 增量采集的数据与转换清洗整个处理链路进行集成及测试 100%
   1. 备注：采集+转换清洗功能已测试，融合数据已经读取转换清洗结果为dataframe；整个处理链路采集到融合输入均已按计划实现
2. 解决融合技术问题：dataframe重构问题
3. 实现融合基于dataframe实现的demo核心功能流程

zhangsiwei6(张思为6) 10-27 18:07:22

10/27

1. 数据融合基于dataframe技术难点讨论解决.

2. 数据融合基于dataframe开发.

3. 添加相应的Test.

数据融合优化 60%

融合基于spark-sql dataframe开发 80%

### 2017/10/28

### 2017/10/29

### 2017/10/30

## Week\_42

### Plan

|  |
| --- |
| 1. 专利书写 2. 车辆接口验证 |

### Require

|  |
| --- |
|  |

### Done

|  |
| --- |
| 增量数据在时间轴上更新图形化分析 |

### 2017/10/16

1. 参加周会
2. 专利资料准备

增量table数据采集功能与任务调度集成 20%

1. vulcanus规则机制理解.

2. udf通用规则开发, 相关文档整理.

3. spark-sql内置functions学习.

UDF开发: 80%

### 2017/10/17

1. 专利书写，完成相关流程图及数据截图

增量table数据采集功能与任务调度集成 30%

10/17

1. UDF通用规则开发.

2. UDF规则开发文档.

3. UDF规则测试.

4. 大数据算法比赛方案讨论.

UDF开发: 90%

### 2017/10/18

1. 专利书写阶段性完成
2. 参加易鲸捷数据库会议
3. 增量table数据采集功能与任务调度集成 40%

### 2017/10/19

1. 参加项目会议

2. 处理案件地图的30w数据

### 2017/10/20

1. 处理案件地图的30w数据按使用方要求转化地区、经纬度、类型
2. 初步处理前端3d功能模块所需数据

* 案件地图数据（李凯）
* 3d功能模块数据（李凯）
* 实体、关系、事件模拟数据（李金龙+张思为）
* 算法（王辉）代码优化（重构）【java、scala】（昨天通锋哥、及辉哥沟通；已取消）

### 2017/10/21

1. 处理案件地图的经纬度, 因为人工百度转换的数据有问题；采用poi库重新处理
2. create temporary function assign as 'com.hikvision.sparta.vulcanus.udf.impl.common.AssignUDF' using jar '/zsw/sparta-vulcanus-udf-assembly.jar';

### 2017/10/22

## Week\_41

* Plan
  + 5s webservice数据接入：重点车辆卡口过车记录数据接入
  + 专利文档书写：一种用于数据管理与认知的数据仓库方案
  + 绩效计划：Q3 绩效计划，Q2绩效计划总结
  + 原有计划开发任务
  + **验证集群化数据有无丢失**
    - **Hdfs 亿级别数据迁移**
    - **Hbase 亿级别数据迁移**

**数据资产的工作：**

**1. 负责db模块，转换清洗模块，webservice模块的开发维护**

**2. 负责统一视图的更新维护；负责围绕在统一视图周围的功能的维护，包括视图解析，规则定义等。**

**3. 负责打包部署模块的维护，主要是编译部署脚本，命令脚本**

**4. 产出对应的markdown文档：vulcanus部署实施手册、vulcanus核心工作、vulcanus统一视图、vulcanus规则机制等**

**1. 数据生成模块开发测试.**

**2. 数据融合模块开发优化测试.**

**3. 服务器各项配置部署.**

**4. 单机版, 集群版CM平台安装配置部署及调试.**

**5. yum代理配置及服务器相关软件安装(oracle,mysql等)**

**6. 出差部署现场环境.**

**7. vulcanus项目熟悉及部署.**

**8. 大数据各个组件知识储备, spark, hdfs, hbase等.**

**9. 相关文档生成.**

**10. 根据算法组的需求对字典表分类整理.**

**11. 根据算法组要求插入数据到218张表,解决相关问题.**

**12. 案件地图经纬度转换.**

**集群**

**Vulcanus小组工作阶段性总结0.7**

* **配合测试同事进行0.6版本测试、专利书写[组内]**
  + **功能、相关知识讲解、缺陷修复**
* **数据资产池相关及识别客户数据更新频率[李凯]**
  + **已做模块基于spark-sql进行开发**
* **集群搭建及数据迁移[张思为]**
* **人脸、车辆本地部署[李金龙]**
* **根据重点id数据接入车辆、人脸数据[李凯]**

**0.6**

* **数据质量，数据资产池试用，大数据公共模块，案件地图地址转换工作[李凯]**

**0.5**

* **数据采集，任务调度，集群梳理[李凯]**
  + **Oracle及hdfs数据采集**

### 2017-10-09

1. 参加周会
2. 初步实现ETL（采集与转换清洗）链路日志重构（data\_version\_load、etl\_core）
   1. ETL链路日志重构（data\_version\_load、etl\_core）50%

<http://10.17.139.67:8080/face-web/ws/rest?_wadl>

./wadl2java -d ../src -p com.hikvision.face <http://10.17.139.67:8080/face-web/ws/rest?_wadl>

C:\ProgramData\Oracle\Java\javapath;%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;%JAVA\_HOME%\bin;%MVN\_HOME%\bin;%HADOOP\_HOME%;E:\developPlat\MinGW\msys\1.0\bin;C:\Program Files\Git\bin;C:\Program Files (x86)\scala\bin;C:\Program Files\nodejs\;C:\Program Files\PostgreSQL\9.4\bin;C:\Program Files\TortoiseSVN\bin;C:\Python33

Error: Registry key 'Software\JavaSoft\Java Runtime Environment'\CurrentVersion'

Unsupported major.minor version 52.0

### 2017-10-10

1. 初步实现ETL（采集与转换清洗）链路日志重构（data\_version\_load、etl\_core）
   1. ETL链路日志重构（data\_version\_load、etl\_core）80%
2. 处理5s webservice技术问题，通过java代码来实际测试人脸的抓拍库查询接口
   1. 解决java url编码问题

### 2017-10-11

1. 数据质量与大数据公共模块基于spark-sql dataframe开发 40%
2. 准备及实施vulcanus组阶段性工作总结讨论
3. 设计webservice数据接入技术实施方案

### 2017-10-12

1. 原有hdfs数据过渡到spark-sql方案设计与开发 60%

c) 实现hdfs文本数据结合mysql元数据写入到spark-sql数据仓库

d) 确定数据质量与大数据公共模块基于spark-SQL开发的具体方案

### 2017-10-13

1. 现场人脸的数据量：79602346
2. 现场的车辆数据接口情况
3. 现场数据的更新频率：通过备份mysql数据库回来
4. 绩效计划填写
5. 思为阅读转换清洗模块，以进行根据spark-sql开发

数据资产统筹规划、维护、开发

1. 原有hdfs数据过渡到spark-sql方案设计与开发 80%
2. ETL链路日志重构（data\_version\_load、etl\_core）80%
3. 数据质量与大数据公共模块基于spark-sql dataframe开发 70%

### 2017-10-14

完成以下模块及通过功能测试

1. 原有hdfs数据过渡到spark-sql方案设计与开发 100%
2. ETL链路日志重构（data\_version\_load、etl\_core）100%
3. 数据质量与大数据公共模块基于spark-sql dataframe开发 100%

## Week\_39

### 2017-09-25

1. 在测试服务器上部署oozie并测试wordcount通过oozie成功
   1. Workflow执行成功
   2. Coordinator执行成功
2. 参加5s对接会议
3. 书写oozie实践文档到组内wiki上

### 2017-09-26

1. 解答测试同事关于oozie任务调度、数据资产池及数据质量的相关问题
2. 设计采集基于spark-sql数据仓库开发的具体方案
3. 进度：
   1. spark-sql数据仓库部署及测试 40%
   2. 采集、转换清洗基于spark-sql数据仓库开发 10%

### 2017-09-27

1. 给测试准备oozie增量数据采集的环境来测试任务调度模块
2. 采集基于spark-sql数据仓库开发 30%
   1. 实现功能逻辑方法基于dataframe处理
3. spark-sql数据仓库部署及测试 50%

### 2017-09-28

1. 给测试准备oozie使用文档,并讲解oozie相关知识
   1. 排查数据质量2张表未进行统计的问题，表属性带有关键字及特殊字符？
2. 采集基于spark-sql数据仓库开发 80%
   1. 实现处理结果封装保存到spark-sql相关的单元功能测试
   2. 实现采集spark-sql功能与原有功能基于输入参数配置可执行
3. spark-sql数据仓库部署及测试 70%

### 2017-09-29

1. 帮助葛挺组在测试服务器上搭建部署案件地图solr搜索引擎
   1. 并将POI数据及已转换百度的地址数据分别写入搜索引擎
2. 转换清洗基于spark-sql数据仓库开发50%
   1. 功能逻辑函数基于dataframe处理
   2. 解析采集load链路日志并实现转换清洗从spark-sql读取数据
3. spark-sql数据仓库部署及测试 80%

### 2017-09-30

1. 采集、转换清洗基于spark-sql数据仓库开发100%
   1. 转换清洗任务处理好的结果数据写入spark-sql数据仓库并将相应的链路日志写入mysql
   2. 并实现转换清洗简单的流程功能测试
2. spark-sql数据仓库部署及测试 100%

1. ssh 到master66

2. spark-sql

3. use vulcanus\_load;

3. select \* from vulcanus\_load.BADPERSON\_S\_FZXYRY;

## Week\_38

## Plan

|  |
| --- |
|  |

### Require

|  |
| --- |
|  |

### 2017-09-18

1. 同智能算法部同事沟通舟山案件地址识别和PIO连接工作
2. 给予测试0.6 数据质量 及 数据质量SQL接口(spark-sql数据仓库)相关文档
3. 实现转换清洗结果数据 数据质量统计

### 2017-09-19

1. 参加压力测试需求评审会议
2. 实现大数据公共模块---Spark-sql数据仓库数据访问接口

### 2017-09-20

1. 完成数据质量--识别源数据更新频率初步设计
   1. 结果mysql数据库表结构设计
   2. 代码处理逻辑流程初步设计
2. 完成测试环境(0.6版本测试)—spark-sql数据仓库部署及验证

### 2017-09-21

1. 协助测试定位解决0.6测试环境大数据环境
   1. 配置更改没重启服务,从而未生效;导致任务cluster模式运行不成功
2. 数据质量--识别客户方数据的条数更新（通过定时进行查询实现）80%
   1. 实现该功能的主要处理逻辑并测试
   2. 实现参数解析模块及调用调度逻辑
3. 大数据公共模块--Spark-sql数据仓库数据访问接口 100%

上次bug描述两个问题：

1. number没有实现，【已实现】

2. 日期类型没有 时分秒 这个，应该在数据库插入oracle数据库的时候，时间数据不对

### 2017-09-22

1. 数据质量--识别客户方数据的条数更新（通过定时进行查询实现）100%
   1. 添加定时调度逻辑，通过quartz实现
   2. 本地ide及master66服务器测试运行通过
   3. 书写部署运行文档
2. 组间联调
   1. 向测试同事通过实际操作讲解yarn相关知识点，及spark job on yarn相关配置

### 2017-09-23

1. 实现如果本次摸底数据与上次摸底数据一致不进行结果保存逻辑
2. 代码review

### 2017-09-24

## Week\_37

### Plan

|  |
| --- |
|  |

### Require

|  |
| --- |
| 1. 完成数据资产组0.7版本的需求和开发计划确定 2. 评估人脸和车辆系统接口可用性，形成技术方案 |

* V0.7需求及开发计划
  + 缺陷修复、代码优化
  + 基于spark-sql开发
    - Spark-sql数据仓库部署及测试
    - 采集、转换清洗、融合基于spark-sql dataframe开发
    - 与数据使用方老葛协商spark-sql使用
    - ETL链路日志重构（data\_version\_load、etl\_core）
    - 原有hdfs数据过渡到spark-sql方案设计与开发
    - 数据质量与大数据公共模块基于spark-sql dataframe开发
    - 上述模块设计及测试
  + 增量table数据采集重构及任务调度集成
    - 增量table数据采集功能重构设计
    - 增量table数据采集功能重构开发
    - 增量table数据采集功能与任务调度集成
    - 增量table数据采集功能与任务调度集成后功能测试
  + 5s人脸及车辆数据接入
* Face\traffic 技术方案
  + 评估人脸和车辆系统接口可用性

### 2017-09-11

1. 深入学习spark-sql技术
   1. 学习spark-sql function模块，了解spark-sql具体有哪些原生function
   2. 学习spark-sql2.2 Cost-Based Optimizer核心概要模型
      1. 通过计算table、column的条数、大小结合计算链路来计算cost

### 2017-09-12

1. 讨论数据资产组内v0.7的工作计划
   1. 具体需要做哪些功能
      1. 具体我和张思为先整理，然后还需要做的邱工补充
   2. 我与张思为分别做哪些功能
   3. 确定v0.7的开发计划任务

### 2017-09-13

1. Intellij 文件头注释模板统一确立
2. Scala 代码单元测试学习及简单实际应用
   * 1. 采用FunSuite风格

数据质量--识别源数据更新频率

data\_load\_pioneer 数据摸底表映射及操作实现

大数据公共模块：

Hdfs 增删封装--hdfs操作核心方法封装

### 2017-09-14

1. 和大家一起初步制定数据资产组v0.7版本的开发计划

a) 具体任务、完成时间及开发人员

2. 学习spark技术

a) 阅读spark 新增主推功能structure streaming官方文档

### 2017-09-15

1. 自己内部运行0.6版本代码,并且就主要功能进行简单功能测试
   1. 修复0.5版本测试提出的缺陷(增量采集未支持Number类型字段,重点测试)

评估人脸和车辆系统接口可用性，形成技术方案。

和锋哥沟通:交通那边没有服务器,给了我们安装文档,我们将他们那套安装在我们的服务器上,(自己造数据);关于数据,看测试那边可有方法解决

### 2017-09-16

### 2017-09-17

## Week\_36

### Plan

|  |
| --- |
|  |

### Require

|  |
| --- |
|  |

### 2017-09-04

### 2017-09-05

1. 部署solr集群
   1. 安装solr集群
   2. 更改solr相关配置并重启生效
   3. 上传solr配置文件并创建zsmap[百度转换的经纬度数据集]、mapzs[poi数据集] 的collection
   4. 将poi数据集写入mapzs collection
2. 数据资产组转换清洗及融合升级
   1. 备份元数据库、清除相关hdfs数据及元数据库表数据及更改相关配置
   2. 更新统一视图并执行转换清洗任务[第二天查看，运行成功(3.5h)]
3. 问题：
   1. 算法python导数据程序内存溢出异常
      1. 通过更改spark-submit参数予以解决

### 2017-09-06

1. 经纬度数据写入solr
   1. 将百度转换的经纬度数据集写入zsmap collection
      1. 问题：程序报 not Serialization Exception
         1. 排查分析：本地运行正常，线上异常；推测为相关maven jar依赖未解决
         2. 解决：笔记本电脑联网解决maven依赖并重新打包
2. 执行融合任务
   1. 执行成功，运行3.5小时
3. 转换清洗、融合结果数据写入spark-sql数据仓库
   1. 删除vulcanus\_core、vulcanus\_merge相关表
   2. 写入转换清洗相关表数据
   3. 写入融合结果相关表数据
4. 执行李金龙vulcanus\_load(贴源层) sql统计语句
5. 帮助葛挺导出vulcanus\_load部分表数据并加密发给葛挺
6. Yarn上自带job，杀不死并且越杀越多
   1. 通过咨询组件组提供解决予以解决[不确定机器重启之后该问题是否重现]

**出差总结：**

* 1. 案件地图poi、百度转换经纬度数据集均按设计导入solr搜索引擎
  2. 数据资产组转换清洗、融合按设计进行升级
  3. 上述出差目标均已达到，并按计划完成

### 2017-09-07

1. 整理出差记录文档
2. 协助算法解决215个卡口7亿多过车数据提取困难
   1. 向邢金彪索要卡口id列表文件
   2. 根据实际需求生成抽数据spark-sql语句
   3. 使用说明：
      1. 
      2. columns 全部替换为实际需要的 列名并逗号分隔
      3. conditions 全部替换为实际需要的where 条件语句 and连接
3. 协助测试理解采集增量实现
4. 表conf\_dict ：为配置字典，
   1. 记录：
   2. 即表示自增规则已配置
5. 表 conf\_dict\_elem ：为字典规则应用
   1. `conf\_dict\_elem\_id` int(11) NOT NULL AUTO\_INCREMENT,
   2. `conf\_dict\_id` 外键 REFERENCES `conf\_dict` (`conf\_dict\_id`)
   3. `elem\_key` 表id
   4. `elem\_value` 属性id
6. 增量实现：在表conf\_dict\_elem中插入记录即可；
   1. Eg：
      1. 解释：即为id=1的源表的attr\_id=1的属性为增量字段
7. 测试问题记录，表属性类型为rowid的采集失败
   1. 已分析(通过走读源码)并测试为rowid底层解析为long，理应解析为string
   2. 解决：通过重写该代码予以解决
   3. 

### 2017-09-08

专利文档书写

1. 修复增量采集数据类型支持bug
   1. 丰富增量字段数据类型支持数学、string、日期等
   2. 测试增量字段数据类型（数学、string、日期）实现等功能[测试通过]
2. 组内工作：书写自己负责模块文档
   1. 学习github wiki使用
   2. 书写 数据质量模块文档

### 2017-09-09

自己负责模块文档书写

专利文档书写

1. 继续书写自己负责模块文档
   1. 书写数据仓库资料文档
      1. 数据仓库部署
      2. 数据仓库访问
      3. 使用数据仓库遇到的问题及解决方法
   2. 案件地图地址转经纬度文档
   3. 案件地图solr搜索引擎部署文档

### 2017-09-10

## Week\_37

# 引用

# 附录

# 0000-年模板

## Week\_00-周模板

### Plan

|  |
| --- |
|  |

### Require

|  |
| --- |
|  |

### Done

|  |
| --- |
|  |

### 0000-01-01

### 0000-01-02

### 0000-01-03

### 0000-01-04

### 0000-01-05

### 0000-01-06

### 0000-01-07