Kafka消费端概要设计说明书

V1.0版本

深圳市蓝泰源信息技术股份有限公司

软件研发部

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修订描述 | 日期 | 作者 |
| 1.0 | 完成初稿 | 2017-01-09 | 杨阳 |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 4](#_Toc471745177)

[1.1. 编写目的 4](#_Toc471745178)

[1.2. 背景 4](#_Toc471745179)

[1.3. 术语和缩略词 4](#_Toc471745180)

[2. 总体设计 4](#_Toc471745181)

[2.1. 运行环境 4](#_Toc471745182)

[2.2. 功能描述和处理流程 4](#_Toc471745183)

[2.3. 存到Redis的key值说明 5](#_Toc471745184)

[2.4. 配置说明 6](#_Toc471745185)

[2.5. 待优化的部分 6](#_Toc471745186)

[3. 运行设计 7](#_Toc471745187)

[3.1. 运行模块组合 7](#_Toc471745188)

[3.2. 运行控制 7](#_Toc471745189)

[3.3. 运行时间 7](#_Toc471745190)

[4. 系统出错处理设计 7](#_Toc471745191)

[4.1. 出错信息 7](#_Toc471745192)

[4.2. 出错处理对策 8](#_Toc471745193)

# 引言

## 编写目的

本说明书编写目的是对Kafka消费端进行概要设计，明确Kafka消费端结构，介绍它的工作过程，运行环境，配置信息。本概要设计说明书的预期读者为app项目组的后端成员以及之后要对Kafka消费端进行扩展和维护的人员。

## 背景

坐公交\_V1.0需要显示的实时车辆信息是坐公交\_V1.0的核心模块。Kafka同时为发布和订阅提供高吞吐量，且可部署为分布式系统，易于向外扩展，为了更快、更好地展示实时信息，采用Kafka对实时信息进行管理。

## 术语和缩略词

**KC —— Kafka Consumer Kafka消费者**

**gw2app —— 网关程序**

# 总体设计

## 运行环境

操作系统：不限

JDK版本：1.8+

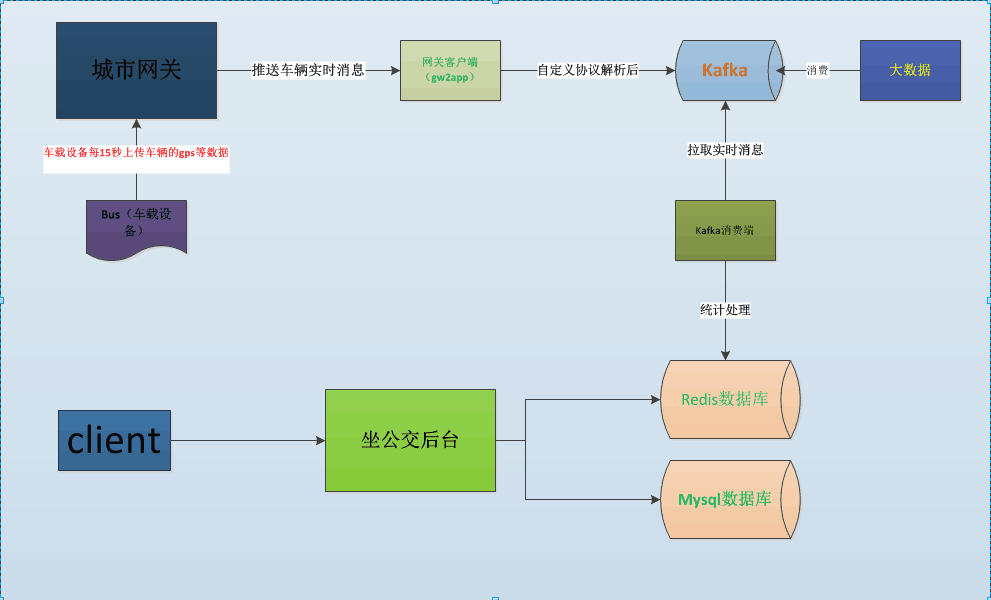
数据库：Redis数据库

## 功能描述和处理流程

每个城市都有一个单独的网关程序（gw2app-cityName.jar）和kafka消费端(KafkaConsumer-cityName.jar)。gw2app负责接收网关实时消息，将消息解析后发布到kafka中。**KC**消费kafka中的车辆实时数据，并将实时数据统计处理后存到Redis中。

不同城市的实时信息发布到不同的topic中，**KC**只会消费自己城市的数据。

详细处理流程如下图所示。



## 存到Redis的key值说明

目前的key值示例：

KEY:PROTOCOLVER:HDBS:PACKETTYPE:gps:GPRSID:101:DIRECTION:0:CITYCODE:440300

“:”是key值之间的分隔符。

PROTOCOLVER:HDBS:网关数据的协议版本。

PACKETTYPE:gps:数据包类型，有gps包（过站包），instation包（到站包），outstation（包）。

GPRSID:101:线路id。

DIRECTION:0:上下行。

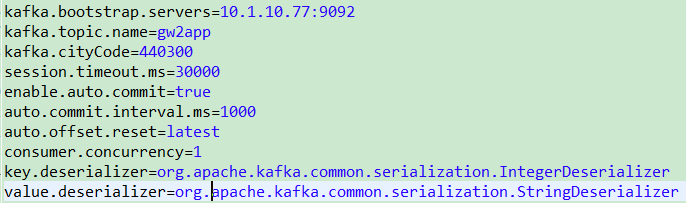
CITYCODE:440300:城市编码。

虽然key值很长，但是有其存在意义。坐公交\_V1.0显示实时信息时，显示的是某条线路上的实时车辆信息，为了快速获取到当前线路上的所有实时车辆，存到Redis的key值需要包含gprsId（线路id，对应坐公交\_V1.0的routeId），direction（上下行）和cityCode(城市编码)，只有这样，才能满足条件。

## 配置说明

不同城市KC的代码都一致，每添加一个新城市，只需要更改两个配置文件即可，分别是kafka.properties和redis.properties。

* kafka.properties配置说明



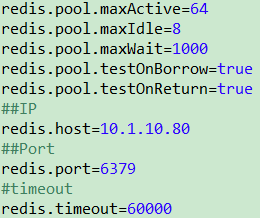
需要更改前三个配置：

kafka.bootstrap.servers：kafka服务端ip和port

kafka.topic.name：要消费的topic（与gw2app中的topic对应）

kafka.cityCode：城市编码

* redis.properties配置说明



需要更改redis.host和redis.port。

## 待优化的部分

现状描述：

1. Redis中存放的key值包含PACKETTYPE，所以每次获取实时信息时，需要获取不同包进行比较，取最新的数据。
2. 所有的数据都是永久存放在Redis中，key值一样的会进行覆盖，坐公交\_V1.0获取实时信息后会查看时间，太旧的信息就舍弃不用。

优化思路：

1. 从key值中去除PACKETTYPE，这样就不用再比较不同包的数据时间，新的数据会覆盖掉旧的，取出来的就是新的数据。
2. 存放数据时，不再永久存放，而是设置一定的expire时间，过期的数据就自动删除。

# 运行设计

## 运行模块组合

KC会被导出为jar包，与gw2app.jar一一对应，每个城市都有一对。只要网关有实时数据发送过来，运行了这两个jar包，坐公交\_V1.0就会显示实时信息。

## 运行控制

后台一直运行，无需控制。

## 运行时间

后台一直运行。

# 系统出错处理设计

## 出错信息

参考logback.xml的配置，出错信息会打印在日志中。

## 出错处理对策

1. 确认Redis与Kafka是否正常运行，不正常的话，需要重启，重启后，再重新启动KC。
2. 查看Redis里是否有实时数据，如果没有，说明gw2app或KC没有正常运行，重启它们；如果有，查看数据的eventTime，可能数据已经太老旧，坐公交\_V1.0不会显示2分钟（配置文件可配置）之前的信息，仍然重启gw2app和KC。