# 软件设计文档

目录

[软件设计文档 1](#_Toc78764074)

[1、概述 2](#_Toc78764075)

[2、总体设计 2](#_Toc78764076)

[3、详细设计 3](#_Toc78764077)

[3.1、模块详细设计 3](#_Toc78764078)

[3.2、算法流程图-时序图 6](#_Toc78764079)

[3.3、接口设计 8](#_Toc78764080)

[3.4、整体设计图 10](#_Toc78764081)

[3.5、数据库表设计 10](#_Toc78764082)

## 1、概述

针对于物流场景的配送时效需求，我们根据提供的数据集进行程序设计，数据集包括揽收时效数据、中转时效数据和派送时效数据。完成接口设计、数据结构设计、算法设计、日志优化、性能测试等一系列工作，满足赛题持续压测30分钟、TP99小于30ms、平均每核CPU承载QPS大于200的需求。并进一步提升性能，远高于赛题要求。

## 2、总体设计

1. 使用Springboot框架进行代码快速实现及轻松部署。
2. 剔除部分不需要的数据，并优化数据类型，保证数据能够存储在内存中。
3. 数据由数据管理类处理，数据结构使用四个map，分别存储揽收时效数据、中转时效数据、派送时效数据、上下级地址映射。
4. 细节优化，如减少对象的创建及日期解析次数，进而减少GC的次数，提升吞吐量。
5. 搜索列表时根据列表大小分为循环和二分两种搜索策略。
6. 默认的Jackson类库改为Fastjson，用于处理请求参数对象及返回参数对象与JSON之间的转换，解析速度稍快。
7. 默认的Tomcat容器改为Jetty，吞吐量提升。
8. 默认的Logback日志框架改为Log4j2，采用全异步策略，根据日志大小进行压缩处理，吞吐量大幅度提升。
9. 虚拟机调优，metaspace设置为100M，避免程序启动时，metaspace初始空间不足，触发多次fullgc。年轻代与老年代的比例从默认的1：2调整到1：1，增大年轻代的空间，可以减少Young GC次数，提升吞吐量，有效利用内存空间。
10. 缓存接口统一，同时实现本地缓存和redis缓存，根据不同的部署场景，配置参数即可切换缓存类型。
11. 数据存到MySQL进行持久化，缓存存储热点数据，确保数据不会丢失。

## 3、详细设计

### 3.1、模块详细设计

**物流时效模块**

**模块描述：**

用户提供下单时间、始发地及目的地等数据，可以获取配送时效。

**模块设计：**

**处理流程：**

1. 程序启动时，创建数据管理中心单例，加载揽收、中转、派送文件到数据结构中。
2. 控制层接收用户传递的参数，然后调用业务层获取时效，业务层首先校验参数合法性，包括下单时间格式、始发省和目的省不为空，各级地址Id为整数等验证。
3. 根据请求参数，从数据中心分别查找对应的揽收、中转、派送时效记录，当任一阶段数据缺失，就返回为空，否则返回时效数据。
4. 控制层根据业务层返回的结果或抛出的异常进行处理，如果业务层处理正常，则返回时效数据则将结果返回给用户，否则将失败的信息返回给用户。

**算法：**

1. 查找时效数据列表：当列表大小<5时，循环搜索，否则二分法搜索。
2. 始发地优先级高于目的地，当目的地未查到对应数据时，则提升至上一级地址，直到目的地为空。当目的地为空时，没有对应数据，则提升始发地地址。

**数据结构：**

1. 存放上下级地址ID映射，key为地址Id，value为上一级地址id

Map<Integer, Integer> upperLevelAddressIdMap

1. 存放揽收时效数据，一级key为始发地Id，二级key为目的地Id，value为揽收时效列表

Map<Integer, Map<Integer, List<CollectInfo>>> collectInfoMap

1. 存放中转时效数据，一级key为始发地Id，二级key为目的地Id，value为中转时效列表

Map<Integer, Map<Integer, List<TransitInfo>>> transitInfoMap

1. 存放派送时效数据，key为目的地Id，value为派送时效列表

Map<Integer, List<DeliverInfo>> deliverInfoMap

1. 揽收时效类：

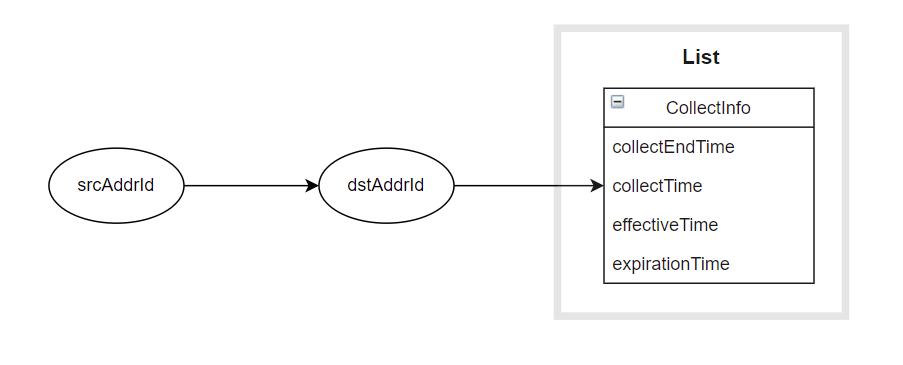
|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 描述 |
| collectEndTime | LocalTime | 揽收截至时间 |
| collectTime | int | 揽收时长 |
| effectiveTime | long | 生效日期 |
| expirationTime | long | 失效日期 |

1. 中转时效类：

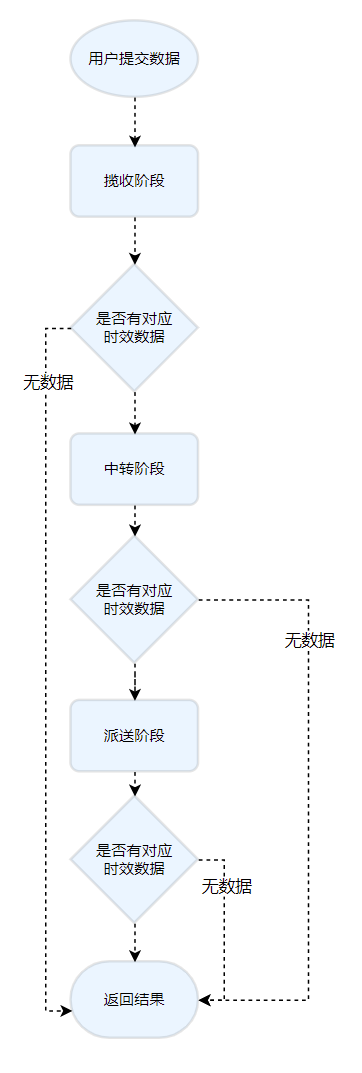
|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 描述 |
| transitDays | int | 中转天数 |
| transitTime | LocalTime | 中转时间（时分） |
| effectiveTime | long | 生效日期 |
| expirationTime | long | 失效日期 |

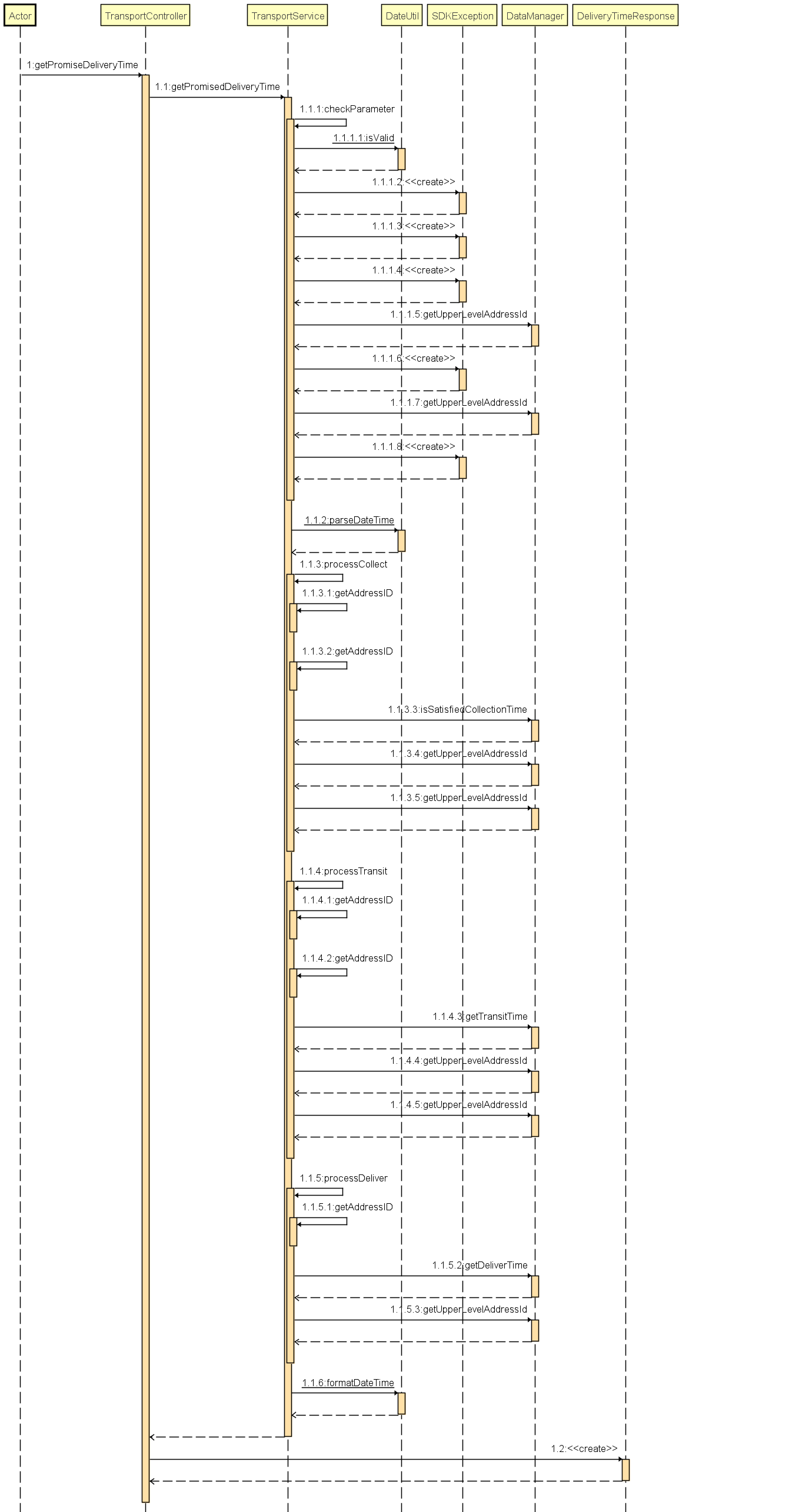
1. 派送时效类：

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 描述 |
| deliverTime | int | 派送时长 |
| effectiveTime | long | 生效日期 |
| expirationTime | long | 失效日期 |



### 3.2、算法流程图-时序图





### 3.3、接口设计

**接口名称：**

查询配送时效

**请求URL：**

<http://localhost:8080/delivery_time>

**请求方式：**

POST

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **必选** | **类型** | **说明** |
| createTime | 是 | string | 下单时间，格式为yyyy-MM-dd HH:mm:ss |
| srcProvinceId | 是 | Int | 始发地省Id |
| srcCityId | 否 | Int | 始发地市Id |
| srcCountyId | 否 | Int | 始发地县Id |
| srcTownId | 否 | Int | 始发地镇Id |
| dstProvinceId | 是 | Int | 目的地省Id |
| dstCityId | 否 | Int | 目的地市Id |
| dstCountyId | 否 | Int | 目的地县Id |
| dstTownId | 否 | Int | 目的地镇Id |

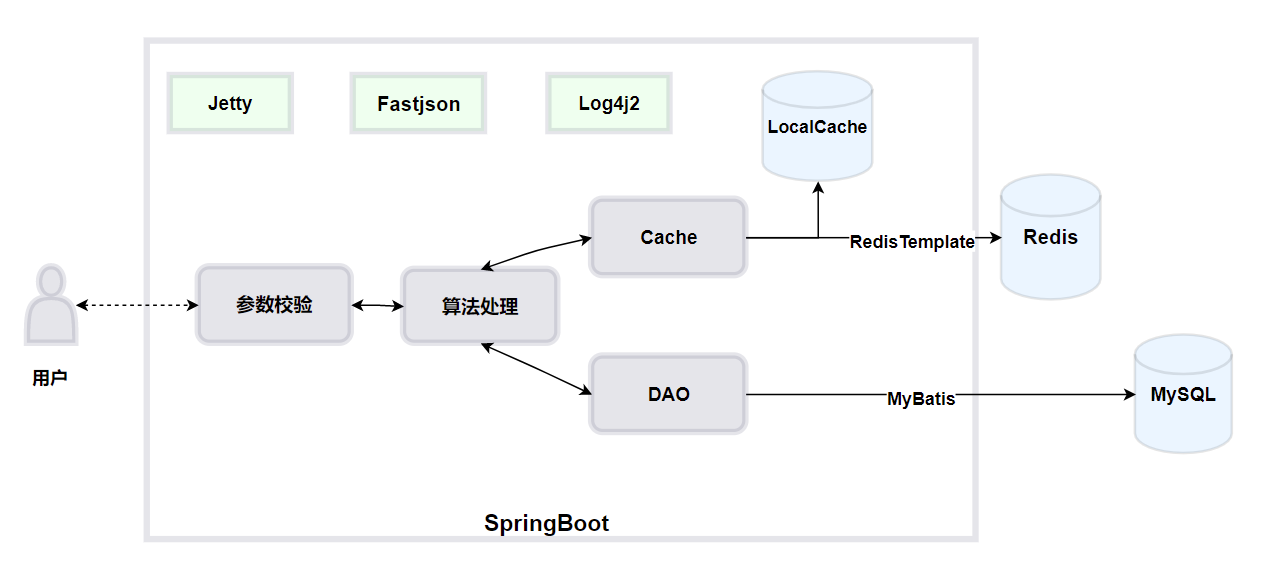
**返回参数说明：**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **说明** |
| code | Int | 错误码，  0：正常 1：异常 |
| time | string | 配送时效数据，错误码为0时，正常显示，错误为1时，不显示该参数 |
| errMsg | string | 错误信息，错误码为1时，显示错误信息，错误码为0时，不显示该参数 |

**返回示例：**

|  |
| --- |
| 正常  {  "code": 0,  "time": "2021-06-23 15:00:00"  }  异常：  {  "code": 1,  "errMsg": "createTime format error"  } |

### 3.4、整体设计图



### 3.5、数据库表设计

1.SQL文件

|  |
| --- |
| drop database if exists trans;  create database trans;  use trans;  drop table if exists t\_collect;  create table t\_collect  (  src\_id int not null comment '始发地id',  dst\_id int not null comment '目的地id',  col\_end\_time time not null comment '揽收截至时间',  col\_time int not null comment '揽收所需时间',  effective\_time bigint not null comment '生效时间',  expiration\_time bigint not null comment '时效时间',  index address\_id (src\_id, dst\_id)  );  drop table if exists t\_transit;  create table t\_transit  (  src\_id int not null comment '始发地id',  dst\_id int not null comment '目的地id',  trans\_days int not null comment '中转所需天数',  trans\_time time not null comment '中转到达时间',  effective\_time bigint not null comment '生效时间',  expiration\_time bigint not null comment '时效时间',  index address\_id (src\_id, dst\_id)  );  drop table if exists t\_deliver;  create table t\_deliver  (  dst\_id int not null comment '目的地id',  deliver\_time int not null comment '派送所需时间',  effective\_time bigint not null comment '生效时间',  expiration\_time bigint not null comment '时效时间',  index address\_id (dst\_id)  );  drop table if exists t\_upper\_addr;  create table t\_upper\_addr  (  id int not null comment '地址id',  upper\_id int not null comment '上级地址id',  index address\_id (id)  ); |

2.表结构

t\_collect表：



t\_transit表



t\_deliver表：



t\_upper\_addr表：

