1. **技术选型**

JDK版本：1.8

非关系型数据库：MongoDB

Http服务器：Redkale

工具类包：Guava

JSON解析工具：FastJson

1. **功能模块**

请求分发：对所有请求进行分发，通过多线程池进行请求的队列管理

对外服务：提供基于JSON数据格式的HTTP协议接口

路由选择：可以自动选择对应服务最佳路由，封装内部链路逻辑

服务管理：对所有服务进行注册，管理，包含流量控制等

监控系统：提供可视化的监控页面，可以人工管理

安全认证：对会话进行有效的安全认证及敏感信息加密

目前开发版本为1.0版，包含各项主要功能，并采用mongoDB存储相应的数据结果进行管理

1. **接口定义**
2. 服务请求接口

请求地址：/services/get

请求方式：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 是否必填 | 默认值 | 说明 |
| serviceCode | 是 |  | 请求的服务编码 |
| identification | 是 |  | 身份识别码 |
| data | 否 | {} | 请求数据，必须使用JSON标准格式 |
| 除此以外的参数将视作传递参数，请求服务时透传 | | | |

返回结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 是否必填 | 默认值 | 说明 |
| statusCode | 是 |  | 结果状态码：  200 已正确接收到请求  300 请求权限不足  400 请求参数错误  500 服务错误 |
| message | 是 |  | 状态码的描述 |
| questId | 是 |  | 请求任务编号，用于获取结果或状态 |

1. HTTP代理发送请求服务接口

请求地址：/proxys/request

请求方式：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 是否必填 | 默认值 | 说明 |
| url | 是 |  | 必须包含协议头的完整url |
| identification | 是 |  | 身份识别码 |
| async | 否 | false | 异步标志，否表示同步接收结果 |
| data | 否 | {} | 请求数据，必须使用JSON标准格式 |
| charset | 否 | UTF8 | 编码集格式 |
| method | 否 | POST | 请求方法，POST或者GET |
| head | 否 |  | 附加文件头，必须使用JSON标准格式 |
| isbody | 否 | false | 是否以Body形式传输数据，true或者false,默认都是参数形式传输数据 |
| businessId | 否 |  | 请求业务的ID，需保持唯一，可用于查询 |

返回结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 是否必填 | 默认值 | 说明 |
| statusCode | 是 |  | 结果状态码：  200 已正确接收到请求  300 请求权限不足  400 请求参数错误  500 服务错误 |
| message | 是 |  | 状态码的描述 |
| questId | 是 |  | 请求任务编号，用于获取结果或状态 |

1. HTTP代理结果查询服务接口

请求地址：/results/get

请求方式：POST

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 是否必填 | 默认值 | 说明 |
| identification | 是 |  | 身份识别码 |
| questId | 两种必填一种 |  | 请求任务编号（来自服务端） |
| businessId |  | 业务ID（来自请求方） |

\*业务ID具有不可信的特点，建议使用服务端凭证

返回结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 是否必填 | 默认值 | 说明 |
| statusCode | 是 |  | 状态码：  200 已获得结果  201 结果尚未返回  300 请求权限不足  400 请求参数错误  500 服务错误 |
| data | 是 |  | 结果数据，无数据时为空集合 |
| message | 是 |  | 状态码的描述 |

1. ESB服务状态查询服务接口

请求地址：/manger/status/get

请求方式：POST

请求参数：无

返回结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 是否必填 | 默认值 | 说明 |
| statusCode | 是 |  | 状态码：  200 正确响应  300 请求权限不足  400 请求参数错误  500 服务错误 |
| data | 是 |  | 结果数据 |
| --queueSize |  |  | --队列中元素个数 |
| message | 是 |  | 状态码的描述 |

1. **存储结构**

||--questId 请求编号

||--url 请求地址

||--url 请求地址

||--status 状态

||--identification 身份识别码

||--async 异步标志

||----charset 字符编码

||----method 请求方式

||----head 请求头部

||----isbody 是否体发送

||----timeout 超时设置

||----businessId 业务编号

||--params 请求参数

||--requestTime 请求时间

||-- excuteTime 执行时间

||--responseTime 响应时间

||--requestIP 请求IP地址

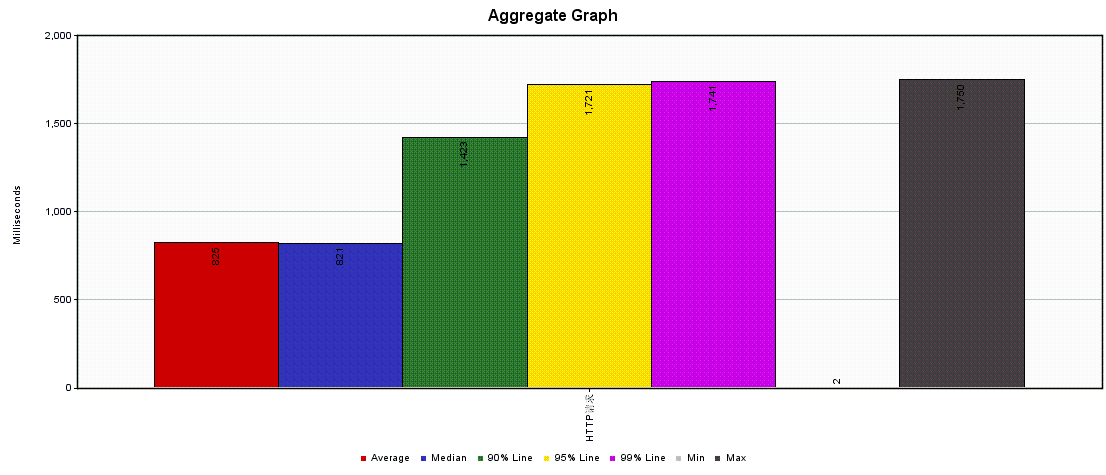
||--result 结果数据

状态码说明：1201已接收；1202已请求；1203已返回； 1400请求失败；

1. **性能指标**

以1秒内发起1000次结果请求来计算，本机测试性能效果如下：

|  |  |
| --- | --- |
| 平均响应时间 | 825ms |
| 响应时间中位数 | 821ms |
| 最短响应时间 | 2ms |
| 最长响应时间 | 1750ms |
| 错误率 | 0% |



1. **业务模块解决方案说明**
2. **POS通转发**

**背景：**目前第三方支付服务端与客户端都是采用Socket连接的方式进行的，当一笔交易进行过程中，连接始终都占用。由于POS通没有支付中这种状态，导致交易时长时间占用Socket连接，多并发时效率低下。

**方案：**采用ESB服务总线的HTTP透传功能，将POS通交易请求转交给ESB进行透传，ESB返回支付中给客户端，客户端确定客户支付完之后，再向ESB请求结果，从而使得每次操作都是短连接，释放资源。

客户端

服务端

ESB

请求

转发

返回

转发

SCOKET

HTTP

**流程：**

1. 客户端发送交易的请求给服务端（Socket通信）
2. 服务端将请求解析后，发送给ESB（HTTP通信）
3. ESB将该请求转发给POS通服务，并详细记录请求和响应，将已经收到请求的信息返回服务端（HTTP通信）
4. 服务端得知发送成功，向客户端发送“支付中”状态（Socket通信）
5. 客户端向服务端请求查询支付结果（Socket通信）
6. 服务端转发请求给ESB（HTTP通信）
7. ESB返回结果给服务端（HTTP通信）
8. 服务端返回结果给客户端（Socket通信）

**改动：**

1. 交易服务端：不再向POS通发起请求，而是交给ESB进行，取GlideNo为业务编号
2. 交易客户端：增加支付中的判断，此时可发起查询请求
3. 查询客户端：原报文末尾加上20位GlideNo
4. 查询服务端：判断报文长度，如果无多出的20位，就按照原来的方式向POS通请求；如果有GlideNo，则向ESB请求，ESB结果库有数据则直接返回，无结果则服务端根据交易号向POS同请求（未传交易号，则通过GlideNo查找）