**SUBWAY业务分析说明书**

（设计文件）

**二零一七年十二月**

**修订历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 日期 | 状态 | 修订人 | 摘要 |
| 1.0 | 2017-12 | C | 吴志明 | 起草 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

状态标识：C – Created A - Added M - Modified D - Deleted

# 业务概述

## 业务背景

当用户量大的时候，可能没办法一个server对接所有的client。为了可以让多个server提供相同的服务已达到负载均衡或者说充分发挥多个主机的运算能力。Subway作为消息代理服务器。把client的请求准确的送达参与服务的server上，达到动态增删server和多渠道处理客户端请求的目的。

## 目的和范围

## 术语和定义

subway:“地铁”服务，以此名字只是为了好理解

## 参考资料

暂无

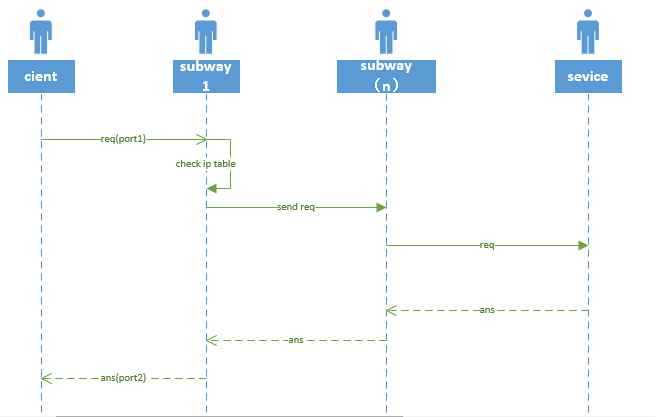
# 需求概述

## 业务要点及设计约束

简单的实时消息队列，发布订阅消息。

# 设计概述

## V1.0系统架构



## 3.2、模块架构

**一、接收模块**

1、一个连接以4K为一个处理单位。每接收4K开一个协程处理。

**二、处理模块**

1、根据消息头定义将消息发送到哪个队列和从哪个队列接收消息。

2、每个队列都存储在内存中。

**三、订阅模块**

1、为每个消息寻找订阅者，找到订阅者后。将消息推送给订阅者。

2、消息存储在固定的消息队列，消息插入和发布都需要上锁。

**四、当前架构思考**

1、经过测试、当数据量大的时候，且如果只有接受端的时候，subway的内存会爆涨。

2、当前代码中，每读取4K开一个协程处理。当数据交互比较大的时候，内存利用率不高。

3、由于消息入队和发布在同一个队列上，在操作队列的时候都需要加锁。使接收模块和订阅模块其实是加密耦合的。

4、消息对包无法排查。

五、当前架构适用性

当前架构只适合在并发量较低且数据量较低，实时的场景