FastIM 框架设计文档

2014-6-21

Michael Tan

本文档描述FastIM项目整体框架设计，针对整个FastIM代码实现过程起指导作用，该设计方案应该与代码实现保持一致。

# 顶层设计

## 全局的部署视图

主要描述用户与服务器的位置关系以及服务性质



## 全局消息的交互协作视图

主要描述客户端与服务器的协作角色、交互情况、消息流向，定义了基本功能。

面对客户端和服务端的IO层时，每一条消息都通过服务器转发，到达目标客户端。



面对客户端平台逻辑层时，要做到的效果是——服务端透明，即客户端的逻辑层按业务封闭数据以后扔给客户端的IO层，由IO层保证正确与服务通信并反馈发送结果。



# 服务器框架设计

## 服务器的用例视图

主要描述服务器需要提供的功能用例



## 服务器实现的构件视图

主要描述通用服务器框架设计，实现层次的划分思想。



LogSystem——日志系统，分三个模块，分别记录IO层、业务逻辑层、数据层的日志；日志提供接口层以方便日志记录功能的改动，应该设计成另外的单独进程，使用管道传输日志信息。

IOLayer——网络接口层，主要负责处理客户端的连接请求和数据双向转输；使用多路分发的Reactor模型来处理，目前打算用成熟的LibEvent库来封装实现。这一层需要设计成完全不处理业务逻辑的功能；在转向业务逻辑层时加适配层，方便记录流量。

LogicalLayer——业务逻辑层，主要负责即时通讯相关的业务处理。业务逻辑层以核心功能代码为主，需要使用IOLayer的发送数据接口以及DataLayer的数据模型接口，但不应该包含直接的网络接口操作或数据层相关的操作——这一层不需要知道IOLayer的数据是怎么来的和怎么去的，也不需要知道DataLayer后面是什么数据库，甚至没有数据库也可以。由于此层业务较多，后续需要再划分细化功能。

DataLayer——数据层，主要负责数据库存储相关的操作，数据层的设计依赖业务逻辑层，为业务逻辑层提供数据模型接口，接口应分层，内层是最小化接口，外层封闭以后是适用业务接口。

# 实现设计

## 服务器各组件的状态机视图

## 服务器各组件之间的交互视图

## 消息数据的活动视图

## 类的静态视图

# 代码设计