MP行情数据处理流程

第一章 MP服务程序的层次结构

MP的服务程序架构

MP的服务程序表现为几个可执行文件（.exe），目前分为32位版本的64位版本，开发环境、测试环境和生产环境中的服务程序都为64位版本，运行于Windows Server 2003 R2 64-bit上，各服务程序需要按照一定的层次结构进行搭建，才能服务MP终端用户的请求。

各服务程序可以运行于不同的PC上，也可以运行于同一台PC上，不过，为了考虑服务程序的性能和稳定性，一般会将其分别部署到不同的PC中。有的服务程序在运行过程中需要占用大量的内存或磁盘空间，因此需要单独部署。考虑到成本、性能、吞吐量等，服务程序的物理部署可以是多样的，但是，它们的逻辑部署是一致的。

关于它们的逻辑部署结构图，可以参考MP文档。

MP主要的服务程序

不同层次的服务程序有着不同的职责。下面，将简单的讲述各服务程序的功能：

（1）Receive Server

该服务程序接收原始行情数据，并进行解析，接着下发。原始行情数据称为Raw Data，它的格式是由交易所规定的，所以Receive Server必须按照交易所给出的协议文档精确的对Raw Data进行解析，同时，在设计Receive Server时也要保证一定的灵活性，因为交易所规定的原始数据格式可能会发生变化。

Receive Server会将原始数据解析成MP内部通用的数据格式，然后下发给它的下层服务程序: Main Server。注意，Receive Server不会对数据进行任何业务处理，它只负责接收->解析->下发。

（2）Main Server

Main Server可以说是MP内部的行情中心，它是Receive Server的下层服务，所有的行情数据经过Receive Server解析之后，都会下发给Main Server，Main Server会根据数据的不同类型进行不同的处理，比如，如果Receive Server下发的是早上的初始化数据（在HKE PRS Spec中对应于DS消息），那么Main Server将会把这些数据拿去进行初始化处理，比如更新Master表等。同时，对于某些行情数据，Main Server会进行一些处理，比如进行业务计算，然后也会把处理之后的结果下发给它的下层服务: Service Server。

（3）Service Server

Receive Server和Main Server可以称为MP的高层服务程序，Service Server及其下层服务被称为MP的低层服务程序，之所以称为低层服务，是因为客户端的请求最终只会到达Service Server进行处理，并不会到达Main Server或更上层的服务，因此，Main Server对于客户端请求时不可见的。所有的请求都会交给Service Server统一处理，当然，如果Service Server不能处理，它会将此次请求转发给能进行处理的服务程序，比如，客户端请求历史数据，而Service Server并不能处理此次请求，但是，它知道DB Inquire能处理，因此，它会将此请求转发给DB Inquire，DB Inquire处理完成之后将结果数据返回给Service Serevr,由Service Server再将结果数据下发以返回给客户端。

（4）DB Inquire

DB Inquire是用于查询历史数据的服务程序，客户端的历史数据请求就是由Service转发给它进行处理，查询出来的结果数据会返回给Service，并由Service下发给客户端。在MP系统中，数据有两个存放地点，一个是内存数据库MDB，一个是物理数据库SQL Server，内存数据库用于存放访问频繁的数据，物理数据库存放大量的历史数据。DB Inquire会根据不同的请求，去不同的数据库中进行查找。

（5）MDB

内存数据库，用于存放可能会被频繁访问的数据，分为MDB\_TICK、MDB\_MINUTELY、MDB\_DAY，MDB\_TICK存放逐笔交易数据，只能存放当天或近期内的交易数据，否则数据量会相当大；MDB\_MINUTELY存放分钟数据，也就是分时图的数据来源，一般也存放近期的数据；MDB\_DAY存放日数据，是日线图的数据来源，几乎存放了从物理数据库有数据以来的所有日数据，也是这三个内存数据库中占用内存最大的，大概1.8G。

（6）DB Save和MDB Save

DB Save是连接到Main Server的，它将Main Server下发给它的数据存放到物理数据库中，比如，Main Server接收到股票的交易行情数据，就发送给DB Save，DB Save就把这一条一条的记录存入对应的Contract表中。

同理，MDB Save会将行情数据存入MDB中，以便防止频繁IO访问带来的性能损伤。

（7）RTS

RTS职责比较简单，但是相当重要。RTS分为内部RTS和外部RTS，外部RTS就是让客户端直接与之相连，而内部RTS是客户端通过Authentication Server分配得到的。不管是内部RTS还是外部RTS，其主要职责是将客户端的请求转发给Service，同时，还要将Service下发的结果反馈给客户端。

（8）Authentication Server

主要负责对客户端的身份进行验证，还有分配内部RTS给客户端进行连接。

第二章 Receive Server处理流程

Receive Server的职责

接收Raw Data，解析，下发。

HKE PRS原始数据的定义

Receive Server的作用是解析原始消息包，不同的交易所，不同的金融品种，其消息包结构都是不同的，因此，要解析原始消息，就必须知道原始消息的具体格式，而这个格式，是在交易所给我们的协议规范中定义的。

我们对Receive Server及其下层所有服务程序的代码分析，都将会以HKE PRS Spec定义的9种不同的消息格式进行讲解。关于PRS原始消息包的具体格式，可以参考《PRS Transmission Spec Version 3.1》。

解析原始消息包

接下来，我们会关注具体的实现代码，来讲解Receive解析原始消息包的流程。

首先，我们要注意两个接口：IPacketAnalyzer和ITransactPacket。

IPacketAnalyzer是框架提供的接口类，其中定义了OnReceiveData方法，任何实现了这个接口的类都必须重写OnReceiveData方法，用于接收原始消息。因此，IPacketAnalyzer实质是用于接收原始行情消息包的接口。注意，IPacketAnalyzer内部有个m\_szPacketData缓冲区（静态数组），用于保存接收到的原始行情消息包。

对于PRS，IPAPRS类实现了IPacketAnalyzer接口，并提供了自己的OnReceiveData方法，用于接收香港的PRS原始行情消息，当接收到了原始行情消息包后，会将行情消息保存于内部缓冲区m\_szPacketData中，接着调用ITransactPacket的DataProcess方法对其进行解析。

ITransactPacket也是框架提供的接口类，其中定义了DataProcess方法，任何实现了这个接口的类都必须重写DataProcess方法，用于解析原始的行情消息，因此，ITransactPacket实质是行情消息解析的接口。

对于PRS，ITPPRS类实现了ITransactPacket接口，并提供了自己的DataProcess方法，专门用于解析香港PRS原始行情消息，同理，ITPSZE类实现了ITransactPacket接口并提供了自己的DataProcess方法，专门用于解析深交所定义的原始行情消息。

ITPPRS内部有一个CCommonData对象的指针m\_pCommonData，指向在堆上的创建的CCommonData对象，该对象内部保存了解析后的数据。

在ITPPRS中，具体的解析任务是由PRSParse对象完成的，HKE PRS Spec定义了9种类型的消息，不同的消息格式是不同的，PRSParse类的职责就是严格按照协议规范解析这9种不同的消息。PRSParse的ParsePacket函数会根据不同的消息类型进行解析，比如如果是DS消息，则会调用ParseDS，如果是TD消息，则调用ParseTD。

以DS消息为例，ParseDS函数内会对原始的DS消息按照协议规范进行解析，并把解析后的数据存储在m\_pCommonData指向的对象中，接着ITP还会将m\_pCommonData指向的数据发往下层服务，即是Main Server。因此，Main Server以及下层所有服务，收到的行情数据都是被放置在CCommonData中已被解析后的数据了。其他原始行情消息的处理过程与此类似。

综上所述，Receive Server的主要两个职责就是：

（1）调用IPAPRS::OnReceiveData接收原始行情消息包

（2） 调用ITPPRS::DataProcess解析消息包，并把解析后的数据下发至Main。

还剩下什么

我们只详细讲解了Receiver Server对HKE PRS的处理流程，对于其他交易所的任何品种，其处理流程都是大致相同的。

同时，关于IPA如何接收数据，ITP如何传输数据，涉及到框架提供的IOCP和线程池。我们只会在业务层面进行探究，关于低层网络通信等代码的具体实现上，可以参考相关书籍自己摸索。

补充说明

Receive Server的业务逻辑代码在ReceivePacket.dll中。

IPAPRS的源码文件路径：

Src\Business\China\ReceivePacket\SocketInterface\IPAPRS.h(IPAPRS.cpp)

ITPPRS的源码文件路径：

Src\Business\China\ReceivePacket\SocketInterface\ITPPTS.h（ITPPRS.cpp）