



A Message System Design Based on Controlled Store-Forward Pattern

汪宏斌 余一清 周洞汝

(武汉大学计算机学院 430072)

摘要: 本论文的课题背景是武汉天河机场的通用机场运营信息管理系统设计与开发。为了保障机场信息系统安全运转,我们设计开发了一个消息处理平台,他是基于可控的存储转发机制模式设计的,通过消息平台,实现生产调度管理信息流的规范化、合理化、科学化。

关键词: 可控 存储转发 消息 代理

1 引言

机场生产运营调度系统的主要通信与信息流是航班动态信息、生产保障动态信息、生产指令信息和各种信息。这些信息具有多对多、实时性强、传递频率高等特征。考虑到在通信过程中,可能会因为各个代理(Agent)间有大量的消息传递和反馈而导致网络发生拥塞,设计中有必要根据整体运作的要求,对各个Agent间的发送与接受关系进行限制,以保障一些主要的Agent间的或

有实时要求的消息传递能正常进行,同时消息的传输与反馈必须有详细的记录,采用可控的消息存储转发机制。对于消息的格式、语义、内容、实现、网络运作、环境、可靠性,采用基于KQML标准,有利于多Agent间的通信。

2 消息传送机制

Agent的消息发送和接收均借助于它内部的消息传递代理对象来完成。网络传输则在相应节点机器上的各个Agent的消息传递代理对象之间进行,消息服务器程序与系统内部的各个子系统之间通过一套完备的、可靠的机制来进行通信,保证消息的传递、接受可靠。假设一次正常的保障, A要发送一条消息给B的一般过程为:

(1) A用户通过本机代理(Agent)发送一条消息到消息服务器(Message Server), A用户继续自己的业务处理。Agent负责处理消息的回执或者重发问题;

(2) 消息服务器接收到A用户的消息后,解析消息报头,识别消息的接收者,并将其存储后

转送给B用户。此时消息服务器日志记录了消息的发送者和发送时间以及接收者和接收时间;

(3) B用户通过代理接收到消息后,立即给消息服务器一个回执(Echo), B用户则根据消息的内容处理自己的业务。

消息服务器在接收到B用户的回执后,给A用户一个回执, A用户的代理接收到回执后,认为这次消息的传送成功完成。

3 消息Agent的实现

系统采用基于定购/发布的消息队列中间件(消息Agent代理)来实现消息流的发布。消息队列中间件采用工业标准的TCP/IP协议,采用了基于Winsock的消息的缓冲存储与基于触发的发布机制与反馈机制,中间件内部还有传输的优先级设定,从而保证消息流的可靠与及时传递。同时,它可以提供定制的消息服务,提高了传递速度并减少了通信开销。通过消息的流动与共享,将各个生产部门紧密的联系在一起,为合理的调度资源,安全、高效的生产给了有力的保障。如

图1所示。

一个本地Agent首先在内部生成一个侦听套接字对象,该对象在程序运行时,自动侦听别的异地Agent发来的连接请求,并为该异地Agent建立一个相应的客户套接字对象。该套接字对象能自动侦测来自该Agent的消息,由本地Agent发往该异地Agent的消息也由该客户套接字对象处理。如果本地Agent向一个尚未建立连接的异地Agent发送消息时,需先建立一个客户套接字,然后由该套接字对象向异地Agent发送连接请求,在连接成功后才能与该Agent进行通信。

从消息源发出的是调度指令消息,从目标返回的消息是应答消息。每个消息都有一个定长的消息头和一个变长的消息体组成,每个消息都以0X0A为结束符,具体格式如表1:

消息帧格式说明如下:

- 消息源指明消息的来源,目的地指明消息的目的地;
- 邮戳日期和时间指明消息源的当前时间;
- 消息类型指明消息是生产指令还是消息应答;
- 消息命令定义消息体的内容;

"I" 插入新纪录;

"U" ... 更新已有记录;

"D" ... 删除记录等;

- 消息体指明具体的消息内容;
- 校验和计算各字节的累加和;
- 结束符消息帧的结束标志。

利用C语言的结构定义,协议结构为:

```
typedef struct {  
    BYTE  Header;    // 帧头  
    BYTE  AddrRes;    // 消息源  
    BYTE  AddrArr;    // 目的地  
    BYTE  MegTime;    // 邮戳日期和时间  
    BYTE  Checksum;   // 校验和  
    BYTE  Type;       // 帧类型  
    YTE  Command;    // 指令类型  
    BYTE  Data[MAX_FRAME_LEN]; // 数据  
    BYTE  Rear  
} GARSOCKETFRAME, *LPGARSOCKETFRAME;
```

4 消息出错处理

消息的传递过程中,可能会出现消息或者回执丢失的情况,为了保障机场的正常运营,我们提供可靠的如下防范机制:

处理一:假设A用户发送给消息服务器的消息丢掉,服务器不会有任何动作,A用户的代理在自己发送消息一段时间T1后,没有收到回执,则重发消息;如果消息再次丢失,仍重发,设定在重发N1次后提醒A用户消息不能到达B用户,由A用户利用其他方式处理这次业务。

处理二:假设消息在服务器传递到B用户的途中丢失,B用户不会有任何动作,消息服务器在自己发送消息一段时间T2后,没有收到回执,则重发消息;如果消息再次丢失,仍重发,设定在重发N2次后发送消息给A用户,提醒A用户消息不能到达B用户,由A用户利用其他方式处理这次业务。若提醒消息在传送给A用户过程中丢失,则成为处理一的处理方式。

处理三:假设B用户收到消息,发送给消息服务器的回执丢失,则消息服务器不能收到回执,消息服务器会以处理二的方式处理。

处理四:假设服务器收到了B用户的回执,记录下B用户的响应时间,然而发送给A用户的回执丢失,则以处理一的方式处理。

通过以上四种处理,我们确保消息无论因何种方式丢失,都不会耽搁业务的保障。

5 结论

通过该系统的建设,实现生产调度管理信息流的规范化、合理化、科学化,实现生产运营中各个工作环节和业务流程的可控性、可追溯性,提高各部门的工作效率与工作质量,提升机场现有的管理水平,通过自身工作的不断改进,充分发挥其整体实力优势,提高机场保障工作的效率,缩短航班停场维护时间,争取最大的航班正点率,获取更高的安全系数,保持同类机场竞争中的领先地位和竞争力。

系统在武汉天河机场运行,证明了系统的实用性,可扩展性以及通用性。 ■

表1 消息帧格式

1	4	4	22	3	1	1	变长	1
帧头	消息源	目的地	邮戳日期和时间	校验和	类型	指令	有效数据	结束符

图1 信息传送流程图

