

【武器装备理论与技术】

基于多 Agent 的某型导弹智能诊断系统中 网络模块的设计与实现

杨 军, 赵 磊

(装甲兵工程学院 兵器工程系, 北京 100072)

摘要:针对基于多 Agent 的某型导弹智能故障诊断系统中的 Agent 之间的通信问题,给出了该系统通信系统总体构成,应用 IOCP 线程管理方式和自定义通信协议数据包,能够实现实时监控状态下的多线程网络通信,最后给出了具体的功能函数定义和整个网络模块中 Agent 间的通信流程。

关键词:导弹系统;故障诊断;IOCP;Agent;数据包

中图分类号:TJ760

文献标识码:A

文章编号:1006-0707(2013)01-0005-04

Network Module Design and Realization of Fault Diagnosis System of One Type of Missile Based on Multi-agent

YANG Jun, ZHAO Lei

(Department of Arms Engineering, Armored Force Engineering Academy, Beijing 100072, China)

Abstract: Aimed at issues of communication in fault diagnosis system of one type of missile based on multi-agent, this paper presented the structure of the communication system of the whole system, application of IOCP thread management mode and custom communication protocol data packets, which can realize real-time monitoring under the multi-threaded network communication, given the specific function definitions and communication among Agents of the entire network module

Key words: missile system; fault diagnosis; IOCP; Agent; data packet

基于多 Agent 的某型车载导弹智能故障诊断系统是利用多个 Agent 对诊断对象进行分布式求解运算而得出故障诊断结果的系统,系统在运行的过程中将涉及到 Agent 之间的通信。所有的通信都是在统一的网络模块下实现的,因此构造一个高效的网络模块对整个诊断系统是至关重要的,网络模块功能的强弱可以影响整个系统故障诊断的效率、速度和系统的智能程度。

1 通信系统总体构成

导弹智能故障诊断系统中 Agent 之间的通信就是指管理 Agent 与各个诊断 Agent 之间的通信。管理 Agent 从故障特征参数获取 Agent 中接收诊断任务之后,就需要与各个诊断 Agent 有效地“沟通”从而完成故障诊断,管理 Agent 与各

诊断 Agent 之间是通过网络模块进行通信,通信系统总体构成如图 1 所示。

图 1 中的网络模块就是 Agent 间通信的工具。每个诊断 Agent 中包含了相应的领域知识库,因此每个独立的诊断 Agent 可以理解为某一领域的专家系统,它所包含的领域知识库一般不相同(也可相同)。管理 Agent 通过任务规划可以找到某一诊断 Agent,利用这个诊断 Agent 自身的分析方法求解出诊断结果,这说明每一个诊断 Agent 都有可能为管理 Agent 提供服务,因此从网络的角度上看,这些诊断 Agent 同时又是一个高性能的服务器,它们可以为管理 Agent 提供网络服务。而管理 Agent 在网络中的角色就是客户端,它可以通过网络模块向各个服务器发出请求并接受来自各个服务器的应答。

收稿日期:2012-09-26

作者简介:杨军(1962—),男,副教授,主要从事导弹工程方面的研究。

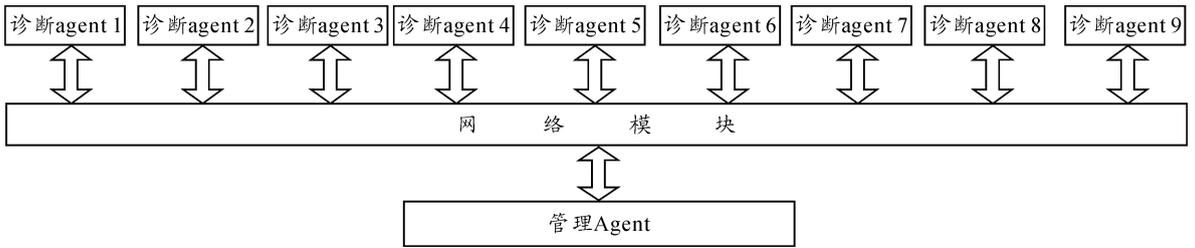


图1 管理 Agent 与诊断 Agent 的通信结构

2 基于 IOCP 的线程管理

由于各个诊断 Agent 都可以看做是一个可以为管理 Agent 提供诊断服务的服务器端应用程序,因此服务器的性能就显得尤为重要,而传统的阻塞同步模式^[1]虽然也可以完成服务器功能,但是这种方法无法处理同时来的多个请求。如果两个客户端同时发出请求,那么只有第一个能得到处理,第二个将等待第一个处理完毕后再进行。Windows NT 3.5 中引入了 IOCP 管理内核对象^[1],IOCP(I/O Completion Port)是一种利用与管理多线程的机制,IOCP 模型是迄今为止性能和可伸缩性方面表现最好的 I/O 模型^[2],因此本系统将构建基于 IOCP 模型的网络模块。

2.1 IOCP 的概念

IOCP 是一个异步 I/O 的 API(Application Programming Interface,即应用程序编程接口),它可以高效地将 I/O 事件通知给应用程序。IOCP 模式要求创建一个 win32IOCP 对象来对异步 I/O 请求进行管理,并通过创建一定数量的工作线程,来为已经完成的异步 I/O 请求提供服务。其次,可以把 IOCP 看成系统维护的一个队列,操作系统把异步 I/O 操作完成的事件通知放入该队列,由于是“操作完成”的事件通知,故取名为“完成端口”,即 IOCP。

2.2 IOCP 线程处理流程

采用 IOCP 模型为服务器端应用程序的网络接口部分提供服务可以获得最佳的系统性能。程序中有两种类型的线程即主线程和工作线程。主线程负责创建并监听套接字(socket),创建工作线程,等待并接受到来的连接,并关联到 IOCP 等;而工作线程则负责等待并处理在 IOCP 对象上完成的 I/O 事件。程序的设计流程如图 2 所示。

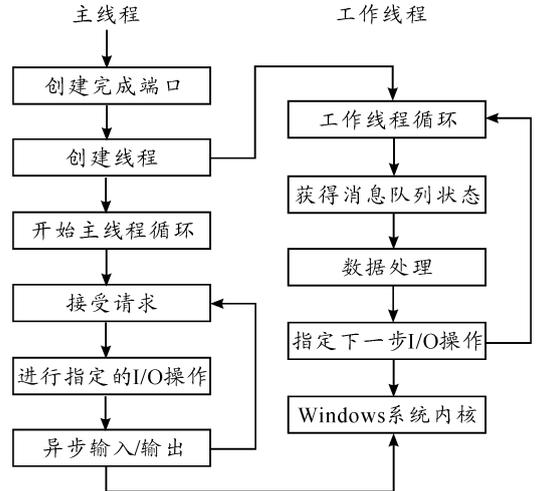


图2 I/O 完成端口处理流程

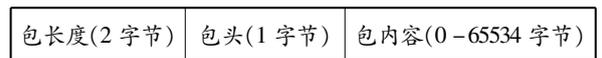


图3 数据包格式

数据包名称定义说明:

从诊断 Agent(DA)到管理 Agent(MA)的数据包为以 ph_DM 开头,从管理 Agent 到诊断 Agent 的数据包为以 ph_MD 开头。

```
enum PACKET_HEAD//包头定义
```

```
{
```

```
ph_MD_Start = 0,
```

ph_MD_QueryBasicInfo, //管理 Agent 向诊断 Agent 发送故障数据的基本信息

ph_MD_AssignTask, //管理 Agent 向指定诊断 Agent 分配任务

```
ph_MD_End,
```

```
ph_DM_Start = 100,
```

ph_DM_BasicInfo, //诊断 Agent 向管理 Agent 发送数据基本信息

```
ph_DM_MarkInfo, //诊断 Agent 返回的属性值
```

ph_DM_Answer, //诊断 Agent 向管理 Agent 返回诊断结果

```
ph_DM_End,
```

```
ph_MD_Count = ph_MD_End,
```

3 通信协议^[3,4]

该系统采用自定义通信协议实现管理 Agent 与诊断 Agent 的通信,数据传递应用数据包的形式进行,下面是 Agent 之间通信数据交换包定义,为减少数据流量,使用二进制包作为数据交换包。数据包的基本格式如下(图 3):

其中包长度指紧跟长度之后的包头和包内容的长度总和。包头用以标记该包的意义,各个包不同。包内容是该包记录的必要信息。

```
};
其中各功能函数定义如下:
1) MD_QueryBasicInfo()函数。该函数实现管理 Agent
向诊断 Agent 发送故障数据的基本信息,其中包括故障任务
编号、任务所处位置信息、任务参数类型、故障特征描述以及
任务优先级等信息。管理 Agent 接收到来自故障特征参数
获取 Agent 的故障任务后,自动生成任务编号和任务分解
后,随即向分解后的一组诊断 Agent 发送这个数据包,其具
体定义如下:
```

```
struct ph_MD_QueryBasicInfo :BasePacket
{
    Int DAID[16]; //参与的诊断 AgentID
    int TaskID; //任务编号
    int Location; //任务所处的位置
    int Type; //任务参数类型
    char Discription[512]; //故障特征描述
    int Priority; //任务的优先级
    MD_QueryBasicInfo() :BasePacket(ph_MD_QueryBa-
sicInfo)
    {
        for (int i = 0; i < sizeof(DAID) / sizeof(int); i +
+ )
        {
            DAID[i] = 0;
        }
        TaskID = 0;
        Location = 0;
        Type = 0;
        Discription[0] = 0;
        Priority = 0;
    }
};
```

2) MD_AssignTask(int DAID)函数。此函数是管理 Agent 在任务分配之后向指定的一个诊断 Agent 发送数据,向此 Agent 下达诊断任务命令,因此此函数传入参数为指定的诊断 Agent 和 ID 编号,其具体实现如下:

```
struct MD_AssignTask :BasePacket
{
    DWORD TaskID; //诊断任务 id
    int questID[ MAX_VALID_QUEST_COUNT]; //故障的
特征描述
    MD_AssignTask(int DAID) :BasePacket(ph_MD_Ass-
signTask) //定义指定分配函数
    {
        TaskID = 0;
        for (int i = 0; i < sizeof(questID) / sizeof(int); i
+ + )
        {
```

```
questID[i] = 0;
}
}
};
```

3) DM_BasicInfo()函数。此函数是诊断 Agent 向管理 Agent 发送基本信息函数,其中基本信息包括该诊断 Agent 位置信息和编号。诊断 Agent 连接到管理 Agent 后同时向管理 Agent 发送该信息。其具体实现如下:

```
struct DM_BasicInfo :BasePacket
{
    int DALocation; //诊断 Agent 的位置信息
    int DAID; //诊断 Agent 的编号
    DM_BasicInfo() :BasePacket(ph_DM_BasicInfo)
    {
        DALocation = 0;
        DAID = 0;
    }
};
```

4) DM_Mark()函数。此函数是诊断 Agent 向管理 Agent 发送的属性值信息,其中包括:诊断 Agent 的活动状态、诊断 Agent 接受过的任务次数、未产生出诊断结果的次数、产生错误结果的次数等信息。管理 Agent 向一组诊断 Agent 发送故障任务后,所有这些诊断 Agent 都向管理 Agent 返回数据包,管理 Agent 将综合这些信息进行任务能力评定以及任务分配。此函数具体实现如下:

```
struct DM_MarkInfo :BasePacket
{
    DWORD TaskID; //诊断任务 id
    int DAActivity; //诊断 Agent 的活动状态
    int DiagnosisTime; //诊断 Agent 接受过的任务次数
    int NoResultTime; //未产生出诊断结果的次数
    int FailedTime; //产生错误结果的次数
    DM_Mark() :BasePacket(ph_DM_Mark)
    {
        TaskID = 0;
        DAActivity = 0;
        DiagnosisTime = 0;
        NoResultTime = 0;
        FailedTime = 0;
    }
};
```

5) DM_Answer()函数。此函数是诊断 Agent 向管理 Agent 发送的诊断结果信息,其中包括任务编号以及诊断结果描述等信息,其具体实现如下:

```
struct DM_Answer :BasePacket
{
    DWORD TaskID; //诊断任务 id
    char questAnswer[512]; //诊断结果描述信息
```

```

double degree; //相似度
DM_Answer() :BasePacket(ph_DM_Answer)
{
    TaskID = 0;
    questAnswer[0] = 0;
    degree = 0.0f;
}
};

```

4 程序实现^[5,6]

网络模块通信具体过程如图4所示。

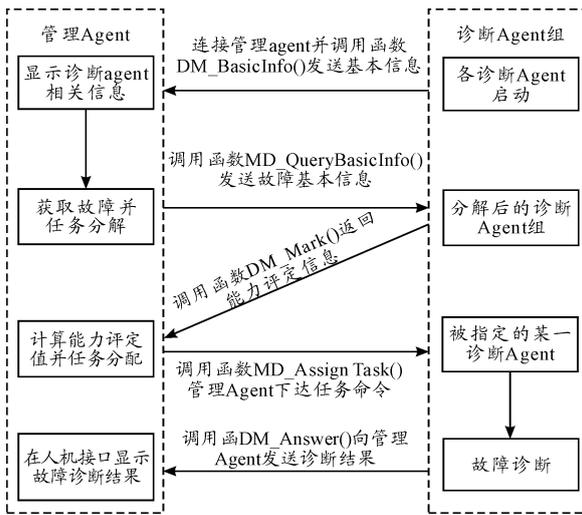


图4 网络模块通信流程示意图

图4所示的是管理 agent 与各诊断 agent 之间通过调用网络模块的功能函数实现通信的过程,系统应用 IOCP 的相关 API 进行线程管理,可以实现管理 agent 同时访问多个诊断 agent,所以图中所示的过程既可以是管理 agent 与多个诊断 agent 同步进行,也可以是管理 agent 与同一诊断 agent 的不同阶段同步进行。针对不同的故障任务每个 agent 进行一次通信之后即进入下一个模块操作,直至故障诊断任务结束。

5 结束语

多 Agent 系统的关键技术之一就在于如何实现多个 Agent 之间的通信协作从而完成某一功能,本系统在建立基于 IOCP 网络构架的基础上,应用自定义通信协议实现了 Agent 之间的通信。应用 IOCP 的管理模式其优点在于它可以满足许多 Agent 同时进行通信而减少不必要的等待,这就大大提高了系统的通信速度和通信效率;在应用层自定义的通信协议能够使系统具有较强的灵活性,使网络模块的可操作性更强。该网络模块实现了实时监控状态下多个故障任务的 agent 之间的通信,应用在该武器系统的故障诊断中解决了系统出现的多故障诊断任务的实时诊断问题,且系统具有可修改性和可扩充性。

参考文献:

- [1] [美] Anthony Jones, Jim Ohlund. WINDOWS 网络编程 [M]. 2 版. 杨和庆, 译. 北京: 清华大学出版社, 2002.
- [2] 王爱平, 唐玄. 基于完成端口的网络服务应用程序开发 [J]. 硅谷, 2009(21): 84.
- [3] Russinovich M E, Solomon D A. 深入解析 Windows 操作系统 [M]. 4 版. 潘爱民, 译. 北京: 电子工业出版社, 2007.
- [4] DOUGLAS E, COMER D, STEVENS L. TCP/IP 网络互联技术(卷3): 客户-服务器编程与应用 (Windows 套接字版) [M]. 张卫, 王能, 译. 北京: 清华大学出版社, 2004.
- [5] 杜翔, 雷跃明. 基于 IOCP 的服务器端应用程序 [J]. 计算机系统应用, 2009(2): 151-154.
- [6] 王艳平, 张越. Windows 网络与通信程序设计 [M]. 北京: 人民邮电出版社, 2006.
- [7] 杨军, 高博. 基于多 Agent 的某型车载导弹智能故障诊断系统 [J]. 四川兵工学报, 2010(8): 4-6.

(责任编辑 周江川)