

航管训练系统中远程控制的设计与实现

刘 佳¹, 胡 术^{1,2}, 莫 晶³, 董志强⁴, 寇成坤¹

- (1. 四川大学 计算机学院, 四川 成都 610064;
2. 四川大学 国家空管自动化系统技术重点实验室, 四川 成都 610045;
3. 民航西南空管局 通信导航监视部, 四川 成都 610202;
4. 空军装备研究院 雷达与电子对抗所, 北京 100039)

摘 要: 航管模拟训练系统是用于培养和提高空中交通区域及近雷达管制人员管制技能的主要手段和标准设备, 能够逼真地模拟包括设备、用户界面等所有与训练相关的对象, 通过实时模拟不同空中交通状态, 以提供一个接近真实的工作环境来训练管制员。文中描述了一个基于可管理线程池的远程自动控制软件 ScriptDispatcher, 并使用 Telnet 协议和 FTP 协议执行配置脚本命令对系统多台主机并发进行程序和数据分发以及远程执行的功能的设计与实现。主要功能包括: 分发训练数据、分发新版程序和远程执行运行脚本等。实践证明, 在大型航管模拟训练系统中, 远程自动控制软件 ScriptDispatcher 可以快速完成训练准备, 运行稳定可靠, 减轻了训练人员的负担, 有效提高了工作效率和系统利用率。

关键词: 航管模拟训练系统; 远程自动控制; 线程池; 数据分发

中图分类号: TP393.09

文献标识码: A

文章编号: 1673-629X(2015)06-0228-06

doi: 10.3969/j.issn.1673-629X.2015.06.051

Design and Implementation of Remote Control in Air Traffic Control Simulator Training System

LIU Jia¹, HU Shu^{1,2}, MO Jing³, DONG Zhi-qiang⁴, KOU Cheng-kun¹

- (1. College of Computer, Sichuan University, Chengdu 610064, China;
2. National Key Laboratory of Air Traffic Control Automation System Technology, Sichuan University, Chengdu 610045, China;
3. CNS Division, SWATMB of CAAC, Chengdu 610202, China;
4. Radar and Electronic Countermeasure Research Institute, Air Force Equipment Academy, Beijing 100039, China)

Abstract: ATC simulation training system is the main mean and standard equipment for training and improving the control skills for air traffic area and approach radar controllers, it can realistically simulate all training related objects including device and user interface, and provide a environment close to the real work to train controllers through simulating the different air traffic status. Describe a remote automatic control software ScriptDispatcher based on a managed thread pool in this paper, which performs configuration script commands to distribute programs and data concurrently to several hosts using Telnet and FTP. And it has completed the design and implementation of remote execution of functions. The main functions include the distribution of training data, the distribution of a new version of the program and remotely executing the running scripts. Practice has proved that the remote automatic control software ScriptDispatcher can quickly complete training preparation, running stably and reliably, reducing the burden of training personnel and improving work efficiency and system utilization in large ATC simulator training system.

Key words: ATC; remote automatic control; thread pool; data distribution

收稿日期: 2014-07-14

修回日期: 2014-10-20

网络出版时间: 2015-05-06

基金项目: 国家自然科学基金资助项目(60832011)

作者简介: 刘 佳(1990-), 男, 硕士研究生, CCF 会员, 研究方向为分布式系统、计算机网络; 胡 术, 通信作者, 博士研究生, 副教授, 研究方向为分布式系统、计算机网络、消息中间件。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150506.1627.017.html>

0 引言

为了适应航管模拟训练系统的训练需求,针对航管模拟训练系统的训练相关的特点,设计并完成了基于航管模拟训练系统的脚本自动分发功能 ScriptDispatcher。该功能通过对与其连接的相关主机进行数据发送,命令传达和文件传输来告知远端主机训练数据的内容,远端执行命令和更新版本程序动作。

1 航管模拟训练系统结构及运行控制功能简介

航管模拟训练系统主要任务是管理空域,制定飞行规则和程序,提供通信导航监视、航行情报、航空气象服务,是用于培养和提高空中交通区域及进近雷达管制人员管制技能的主要手段和标准设备,能够逼真地模拟包括设备、用户界面等所有与训练相关的对象,

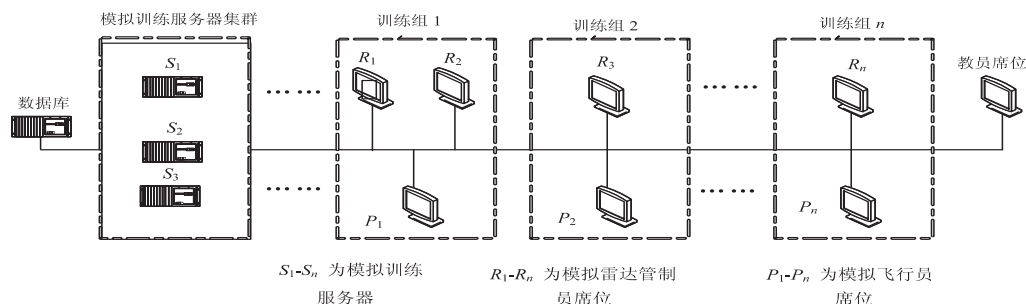


图1 航管模拟训练系统组成图

系统设计时各训练组的进程(含服务进程、模拟机长和模拟管制员席位)均部署相同的进程,如果训练版本需要更新,则使用系统分发新版本程序和远程执行运行脚本的功能进行版本升级。可以针对不同的主机发送不同的训练数据,这可以在任务脚本的配置文件中设置。这样一个教员席就控制了整个训练系统的训练数据的分发和计划的执行,便于各训练组之间的协调,也便于及时发现训练中的问题。

2 ScriptDispatcher 的功能与设计

ScriptDispatcher 就安装在教员席上,作为客户端与远端的训练组主机进行 Telnet 和 FTP 通信的功能软件。它设计的主要目的是为了避免通过人工在具有相同功能和规定动作的机器群中做相同的安装动作或者脚本执行。将机器群与该系统进行网络连接,而后就可以通过系统向机器群中的每台机器添加任务或者发送训练数据。

具体来说,ScriptDispatcher 的 3 个主要功能如下:

(1)分发训练数据。向远端机器发送训练信息。可以指定具体的机器,而不必是一个机器群。这通过传送一个包含训练数据的文件来实现。

(2)分发新版本程序。通过 FTP 的文件传输功能

通过实时模拟不同空中交通状态,以提供一个接近真实的工作环境来训练管制员,并且提供必要的手段完成这一模拟训练过程的准备、运行、记录、回放以及评估等^[1]。

如图 1 所示,在航管模拟训练系统中教员席位用于开启训练过程,准备和生成训练数据,读取本地的任务脚本文件,对参训的模拟机长席位和模拟雷达管制席位进行远程配置,完成一个模拟机长席位和一个或多个模拟雷达管制席位的组队训练。模拟训练服务器集群中部署相同的进程,运行一组或多组训练服务程序,这些训练服务程序由教员席位在开启训练组时启动,在训练组关闭时退出相应组的训练服务程序。教员席位通过启动训练服务进程、模拟机长和模拟管制员席位完成训练组的开启,在训练完成后退出相应的训练组进程。

向远端机器发布新版本,执行并覆盖原来的旧版本程序。

(3)远程执行运行脚本。通过 Telnet 的发送命令功能向远端机器发送远程执行运行脚本的命令来执行脚本。

另外 ScriptDispatcher 还具有垃圾文件删除和 Core 文件收集的功能。

2.1 功能设计

如图 2 所示,ScriptDispatcher 通过读取指定的 XML 格式的任务脚本,获取任务内容和远端主机地址,通过 FTP 和 Telnet 网络库对远端主机进行相应的操作,并在线程池的控制下保持 CPU 资源的有效合理利用。

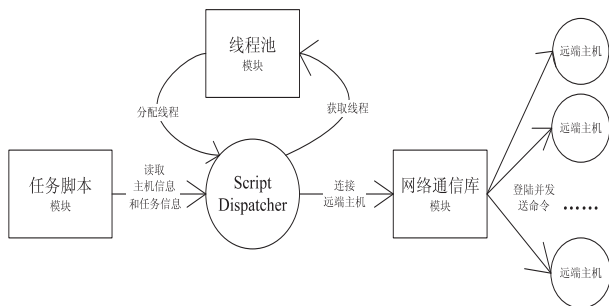


图2 航管模拟训练系统设计图

ScriptDispatcher 的封装库支持 Windows2000 以上的 32 位版本和 Unix 的主流 32 位版本。在 Unix 操作系统中,采用界面化的设计来加强与使用人员的交互性。界面编写使用跨平台语言 Qt。

下面详细介绍 ScriptDispatcher 的 3 个组成模块:任务脚本、网络通信库和线程池。

2.2 任务脚本

2.2.1 任务脚本的定义

任务脚本是记录远端主机的任务信息和主机信息的配置文件。在脚本语言的选择上,Unix 的 Expect 继承了 TCL 的语法规则,虽然内容丰富,功能齐全,但理解起来有一些难度。为了使软件操作的培训更为简单,任务描述更加清晰,这里采用了使用普遍的 XML 来编写配置文件。

2.2.2 XML 语言描述的任务脚本

XML 语言描述的任务脚本有如下的格式:

```
<FTP>
<user>lkl</user>
<password>lkl7890</password>
<mode>I</mode>
<srcpath>../file/168</srcpath>
<group>
<hostinfo>
<ip>168.1.3.61</ip>
<ip>168.1.3.62</ip>
</hostinfo>
<taskinfo>
<task>
<cmd>STOR</cmd>
<arg>hzq.zip</arg>
</task>
<task>
<cmd>TELNET</cmd>
<arg>unzip -o hzq.zip >/dev/null</arg>
</task>
<task>
<cmd>TELNET</cmd>
<arg>&rm -fr hzq.zip</arg>
</task>
</taskinfo>
</group>
</FTP>
```

这个脚本要求 ScriptDispatcher 以用户名 (user) lkl, 密码 (password) lkl7890 登录 IP 地址 (ip) 为 168.1.3.61 和 168.1.3.62 的主机 (hostinfo), 并设置 FTP 的传输模式 (mode) 为 I, 即图像传输。将本地文件作为源路径文件 (srcpath), 作为上传的源文件或者下载的保存文件。执行一组任务 (taskinfo), 具体任务 (task)

为:

(1) 执行命令 (cmd) STOR (Storage), 命令参数 (arg) 为 hzq.zip。在 FTP 中, STOR 命令是存 (储) 放一个文件, 所以这条任务可以翻译为: 在当前目录下将源文件 (168) 存放为名为 hzq.zip 的文件。接着 ScriptDispatcher 调用 FTP 库, 与远端主机进行 FTP 通信, 将目标文件保存为 hzq.zip。

(2) 执行 TELNET 命令, 命令参数为 unzip -o hzq.zip >/dev/null。TELNET 命令是 ScriptDispatcher 自定义的对远端主机的一些操作命令。这条命令的参数是解压 (unzip) 远端主机当前目录下的 hzq.zip 文件到 (>)/dev/目录下的 null 文件中, -o 参数指定如果存在名称为 null 的文件, 覆盖它, 并且不提示覆盖动作。为了执行这条命令, 系统将调用 Telnet 库, 对远端主机进行操作。

(3) 执行命令 TELNET, 命令参数为 &rm -fr hzq.zip, -fr 说明移除 (rm) 动作是强制 (f) 的, 并且递归 (r) 移除目录下的内容。& 符号告知远端主机将此任务作为后台任务运行。

任务脚本的描述很简单, 格式也很统一, 甚至可以将一类任务写成模板的形式, 只需要修改其中的 hostinfo 标签信息和 srcpath 标签信息, 所以也很方便。

3 网络通信库的实现

3.1 Telnet 功能实现

随着因特网基础设施被越来越多地用于商业应用, 对服务质量的需求似乎成为越来越多的网络服务的基石^[2]。Telnet 全称电信网络协议 (Telecommunication network protocol), 起源于 1969 年的 ARPANET^[3]。Telnet 是使用最频繁的工具, 各种分析所用的数据基本都可以通过 Telnet 到设备^[4]。Telnet 协议可以工作在任何主机或任何终端之间, 因此采用 Telnet 协议来向远端主机发送命令。Telnet 通信两端必须把自己的终端特性转换为统一的网络虚拟终端^[5]。

Telnet 协议定义了标准接口, 可用于实现界面终端和面向终端的过程并实现很好的交互, 以及终端到终端的连接, 将来还非常有可能用于实现处理与处理即分布式计算, 这都归功于其通用的、双向的、面向八字节的通信方法^[6]。

Telnet 客户端的工作流程为:

- (1) 请求远端主机建立 Telnet 连接;
- (2) 进行选项协商和子选项协商;
- (3) 输入用户名和密码进行登录;
- (4) 发送命令;
- (5) 关闭连接。

编写 Telnetlib++ 作为 ScriptDispatcher 与远端主机

间的 Telnet 通信库。以下是库在设计时做的一些处理:

(1)对选项协商的处理。Telnet 通信的两个方向都采用带内信令的方式。为了简单起见,在处理选项协商或者子选项协商时,本地所能接受的选项是很有几个,只处理了一些重要的命令,如 IAC (Interpret As Command,作为命令来解释,连续两个的 IAC 表示数据字节 255),DM (Data Mark,数据标记),SB (Sub Begin,子选项开始),SE (Sub End,子选项结束),DO (激活选项),DON'T (禁止选项),WILL (同意)和 WONT (反对)。为了减少处理的复杂程度,对于一些不常用的选项,如 EC (转义字符),EL (删除行)等,以及一些异常处理的选项如 AO (异常终止输出)没有做正确处理,而是直接回复 WONT。一来由于系统的操作有限,出现以上的不常用的选项的概率非常小;二来异常情况很少发生,即使发生也可以通过系统的其他模块检测出来。而且也没有主动要求对端主机设置一些选项。实践证明,这种简化并没有影响系统的正常工作,而且效率也是很高的。

(2)与远端主机即服务器的通信的处理。Telnetlib++是这样处理服务器发送给系统消息的:首先将收到的信息串解析,如果有 IAC 并且不是连续的 IAC,说明这是与服务器的协商,解析出这次服务器发给本地的 Option,加入到 Option 的链表中,并将其余的信息进行递归解析。然后根据上面列出的几个重要的选项对这些 Options 做出同意或反对,激活或禁止的回应,并显示信息串的数据内容。否则,这是远端主机对命令的回应,例如回应 ls 命令给出文件列表,将这些显示到屏幕或者记录到文件中。而系统发送给远端主机的信息只需要通过 TCP 向远端主机指定的端口发送即可。

3.2 FTP 功能实现

随着互联网的快速发展,文件共享成为互联网不可或缺的功能之一。文件共享方法也是多种多样,如 FTP 协议。FTP 也叫文件传输协议,是英文 File Transfer Protocol 的缩写。FTP 是一种应用层协议,是 Internet 中最早提供的服务之一,也是最为普遍的网络协议之一。FTP 协议采用现在比较流行的 C/S 模式实现文件的传输,它由 FTP 客户端程序和服务器端程序构成。用于文件传输的 Internet 标准^[7-11]。为了建立与远程系统的连接,从保密安全的角度出发,FTP 要求客户向服务器提供用户注册名和口令,服务器拒绝非法用户的访问。但是连接一旦建立成功,一个或多个文本或图像二进制文件都能被传送,FTP 不必担心可靠性和连接的管理,因为 FTP 依靠 TCP 正确执行这些功能^[12]。

系统使用 Ftplib++作为本地客户端与远地主机的

FTP 连接所使用的库。Ftplib++是 FTP 客户端的一个简化版本。它包含了一些常用的 FTP 操作,而且简化了对 FTP Server 应答的处理。

Ftplib++并不关心具体的 server 应答是什么含义,只是想了解这个操作是否成功。如表 1 所示,server 的应答以数字 1~5 开头,所以 Ftplib++在发送完命令后会检查 response 的开头一位是不是与预期相符,如果不符,则表示命令失败了。例如登录时,在发送 USER lkl\r\n\0 后,会期待得到首位为 3 的应答来输入密码,如果不是 3,则表示登录失败。类似地,发送一个 MKDIR 命令后期望得到首位为 2 的回复,否则说明 dir 并没有被创建。

表 1 FTP Server 应答首位含义

应答	说 明
1yz	肯定预备应答,它仅仅是在发送另一个命令前期待另一个应答时启动
2yz	肯定完成应答。一个新命令可以发送
3yz	肯定中介应答。该命令已被接收,但另一个命令必须被发送
4yz	暂态否定完成应答
5yz	永久性否定完成应答。命令不被接收,并且不再重试

FTP 客户端的工作流程为:

- (1)与远端主机建立 FTP 连接;
- (2)输入用户名和密码进行登录;
- (3)传输文件;
- (4)断开连接。

ScriptDispatcher 使用 Ftplib++实现上述的客户端动作。下面详细介绍 Ftplib++。

连接和登录。在与远端主机建立 FTP 连接时,如果没有指定特定端口,就连接主机的 FTP 端口。通过设置 socket 的 SO_REUSEADDR 选项来指定地址重用。

Ftplib++的优点。Ftplib++除了支持 RMDIR,CHDIR,PUT,GET 等常用的 FTP 命令外(这些命令都是通过直接向 FTP 服务器发送命令信息来实现的),还支持 PUTA,GETA 的操作,A 的意思是 Append,也就是在目标文件追加读写。并可以指定源文件的读取和写入位置。这给系统的工作带来了很大的方便。另外,如果 PUT 和 GET 的参数类型不是一个普通文件而是一个路径名,就会将源路径的所有文件都拷贝到目的路径中。这也使操作更加简捷。但在传送之前,还必须确定传输的路径或者文件名是否存在,如果不存在则创建。另外 Ftplib++还拥有 resuming,fxp,ssl/tls,encryption 等功能,但是这功能并没有在系统中用到。

Ftplib++还可以指定两个回调函数:log 回调,格式为 typedef void(* FTPCallbackLog)(char * str, void *

arg);idle 回调,格式为 typedef int(* FTPCallbackIdle)(int xfered, void * arg)。log 回调在登录失败时调用,参数可以自己指定,例如可以将输入的用户名密码反馈给客户提醒其更正或者输入匿名用户再次进行登录。而 idle 回调在数据连接在 FTPRead 和 FTPWrite 的过程中被调用,调用 Ftplib++的程序可以根据所传输的字节数计算该文件的传输进度,或者在连接空闲时处理其他的事务(当接收到 0 字节时)。

控制连接和数据连接的连接时机。客户端的操作有两种。一种是发送命令,在远端主机执行,取得操作的返回码。另一种是发送命令后,服务器通过数据连接传送数据。这就需要建立数据连接。尽管 PASV 模式可以避开对防火墙的设置,但是为了安全起见,可能需要在主动模式下打开数据连接。当类似 GET 和 PUT 这样的操作执行时,都会对数据连接的类型和生命周期进行检查,如果类型不符合或者连接断开,就要重新创建新的数据连接。

Ftplib++并没有实现暂停和断点续传功能。

3.3 Telnet 和 FTP 的联合工作

系统支持的操作有:

```
const int FTP_SEND=1; /* 上传文件 */
const int FTP_GET=2; /* 下载文件 */
const int FTP_CD=3; /* 改变远程目录 */
const int FTP_MKDIR=4; /* 创建远程目录 */
const int FTP_SITE=5; /* 发送 SITE 指定的命令 */

const int FTP_LCD=6; /* 改变当前的目录 */
const int FTP_TELNET=7; /* TELNET 命令 */
```

其中,1~5 都是通过 Ftplib++来实现的,这对应系统分发训练数据和分发新版本程序的功能,FTP_LCD 是通过执行本地的 system 函数来进行的,FTP_TELNET 是通过 Telnetlib++实现的,这对应系统远程执行运行脚本的功能。两者在系统中的运行流程为如图 3 所示。

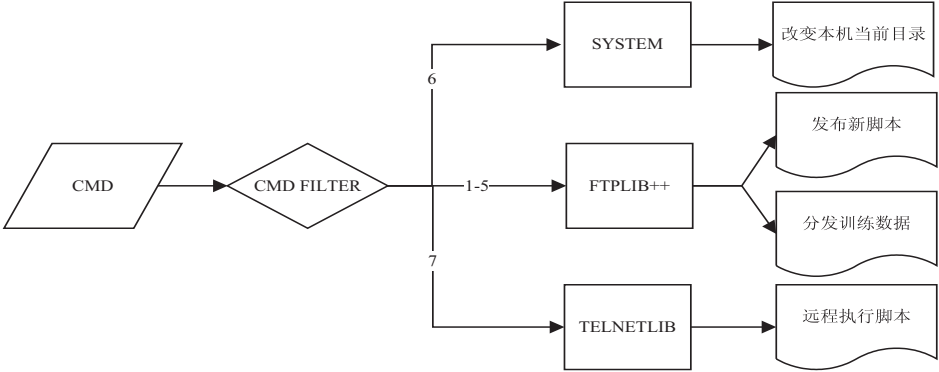


图 3 网络库的工作流程

4 基于线程池的数据分发和执行

4.1 线程和线程池策略

Expect 并不能在同一个界面上同时对多台主机进行 Telnet 或者 FTP 连接,ScriptDispatcher 则可以在一个界面上对多个主机发送不同的命令,这主要得益于它的线程和线程池策略。

同时,能够采用线程策略,也得益于设计良好的 Telnetlib++和 Ftplib++网络库,它们都是线程安全的,为多线程并发执行提供了可能。

采用线程主要为了避免在一个远程主机没有应答的时候,系统一直处于等待该主机应答的状态,而耽误了对其他主机的操作,充分发挥多处理器的优势。采用线程池,主要是由于基于线程的创建和销毁浪费太多的 CPU 资源。因为对每台远端主机的连接可能会很频繁地发起和放弃。

基于“One Event Loop One Thread”的线程设计原则,系统采用一台主机一个线程的策略。这样可以

不同主机的操作隔离开来不会形成运行依赖。粒度不适宜太小,没有必要对 Telnet 和 FTP 分建一个线程。

加入线程池的线程被称为池线程。如图 4 所示,在线程池的层面上,池线程从一开始创建到最后销毁,只有两种状态:阻塞态和运行态。线程池预先创建一些线程,这些线程在没有被调用的时候一直处于阻塞状态(通过调用 pthread 的 Wait 实现),并被加入线程池的 free_list 链表。当需要分配线程的时候,线程池从 free_list 链表中取出一个线程(通过调用 Signal 函数来唤醒等待中的线程),在唤醒线程之前为其注入任务(start 动作,run 动作以及 end 动作),这样线程就会进入运行态,依次执行这三个动作,完成后,线程又被加入到 free_list 链表,调用 Wait 进入阻塞态,等待线程池的唤醒。

4.2 远程执行可能失效下的处理策略

在远端主机 CPU 占用率很高,或者命令需要耗费远端主机大量资源,或者远端主机突然死机的情况下,系统的命令可能会延时相当长的时间或者永远得不到

执行,此时需要终结掉该线程。如在 Linux 系统下调用 pthread_cancel 函数,设置异常退出函数^[13],采用

pthread_cleanup 函数进行异常处理,释放之前加锁的互斥锁、内存等资源。

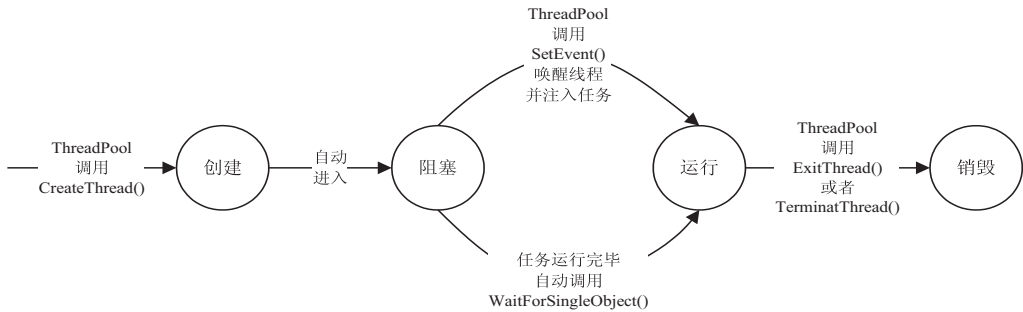


图4 池线程的状态转换

但是由于线程的调用会很频繁,所以大量调用入栈出栈操作的耗费与重启程序所造成的浪费需要被很好地衡量。为了避免这种情况,在用户认为这是一个资源耗费很大的操作时,可以采用 Cancel 模式,而平时只采用一般的等待线程退出的模式。

4.3 分发进度的图形表示

虽然 ScriptDispatcher 能在控制台下运行,但是为了交互方便,ScriptDispatcher 提供了人机交互界面,如图5所示。

Qt 提供了一个 QProgressBar 图形界面来表示操作的完成情况。采用这个控件来表示命令的完成情况:每个线程的任务完成情况之和的平均数,即为总的完成情况。结合 QTimerEvent,可以定时刷新任务完成情况。

在航管训练系统的后续研发中,随着训练功能的日益丰富,训练数据的日益增加,训练反应要求的日益提高,对脚本自动分发的速度和可靠性提出了更高的要求,其 XML 任务列表的操作内容也会不断丰富,任务主机的数量也会不断增加。这就要求采用更为快速的文件传输方式来进行两机间的通信,并采用有效的线程池管理更加有效地安排任务。由于篇幅所限,这里不再赘述。

参考文献:

[1] 谢进一,石丽娜.空中交通管理基础[M].北京:清华大学出版社,2012.

[2] Kumar A, Karthikeyan S. Security model for TCP/IP protocol suite[J]. Journal of Advances in Information Technology, 2011,2(2):87-91.

[3] Steven W R. TCP/IP 详解(卷一)[M].北京:机械工业出版社,2008.

[4] 马坤,杨波. Telnet 协议在网管系统中的应用[J]. 信息技术与信息化,2006(4):125-126.

[5] 卢爱卿,张会勇,赵征. Telnet 协议的实现原理及应用[J]. 计算机工程,2002,28(11):268-269.

[6] 林勇.基于 SNMP 和 TELNET 协议的 IP 承载网智能化维护系统设计与实现[D].成都:电子科技大学,2013.

[7] Postel J. File transfer protocol[S]. RFC 959,1985.

[8] 李柳,吴丽娟,王占军. 分布式 FTP 系统的设计与实现[J]. 辽宁工程技术大学学报:自然科学版,2005,24(2):232-234.

[9] 许君,王朝坤,李瑞,等. 基于内容的分布式 FTP 搜索引擎的设计与实现[J]. 计算机研究与发展,2011,48(S):430-435.

[10] 周宏生. 基于 FTP 协议的文件上传和下载[J]. 科技资讯, 2006(2):5-6.

[11] 张葵,李鹏. (四)如何在 Internet 上传输文件[J]. 电子科技导报,1998(8):30-34.

[12] 黄世权. FTP 协议分析和安全研究[J]. 微计算机信息, 2008(6):93-94.

[13] Steven W R. UNIX 网络编程(卷二)进程间通信[M].北京:人民邮电出版社,2010.

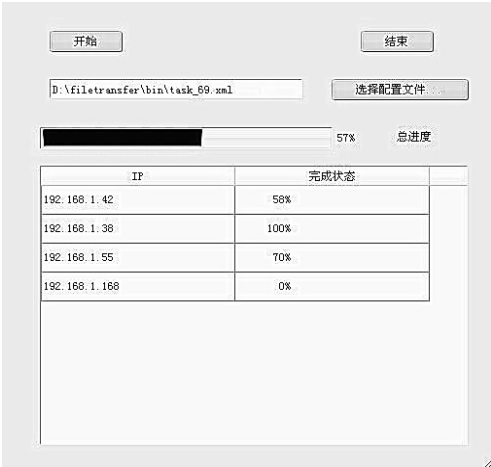


图5 系统运行的界面

5 结束语

文中介绍了一种脚本自动分发执行的功能 Script-Dispatcher,使用 XML 描述任务,ScriptDispatcher 从文件中读取任务加入到任务列表,根据任务的性质,使用 FTP 和 Telnet 进行远端控制和传输,并在线程池的控制下有效提高系统利用率和工作效率,避免了对多机进行繁琐的相同操作,减轻了训练人员的负担。

作者：[刘佳](#)，[胡术](#)，[莫晶](#)，[董志强](#)，[寇成坤](#)，[LIU Jia](#)，[HU Shu](#)，[MO Jing](#)，[DONG Zhi-qiang](#)，[KOU Cheng-kun](#)

作者单位：[刘佳,寇成坤,LIU Jia,KOU Cheng-kun\(四川大学 计算机学院,四川 成都,610064\)](#)，[胡术,HU Shu\(四川大学 计算机学院,四川 成都 610064; 四川大学 国家空管自动化系统技术重点实验室,四川 成都 610045\)](#)，[莫晶,MO Jing\(民航西南空管局 通信导航监视部,四川 成都,610202\)](#)，[董志强,DONG Zhi-qiang\(空军装备研究院 雷达与电子对抗所,北京,100039\)](#)

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2015 (6)

引用本文格式：[刘佳](#).[胡术](#).[莫晶](#).[董志强](#).[寇成坤](#).[LIU Jia](#).[HU Shu](#).[MO Jing](#).[DONG Zhi-qiang](#).[KOU Cheng-kun](#) [航管训练系统中远程控制的设计与实现](#)[期刊论文]-[计算机技术与发展](#) 2015 (6)