

引用格式:周敏刚,牟明,王泉,等.综合模块化航空电子软件分布式测试平台研究[J].电光与控制,2018,25(2):74-78. ZHOU M G, MU M, WANG Q, et al. Distributed testing platform for integrated modular avionics software[J]. Electronics Optics & Control, 2018, 25(2):74-78.

综合模块化航空电子软件分布式测试平台研究

周敏刚, 牟明, 王泉, 张蕾

(中国航空工业集团公司西安航空计算技术研究所,西安 710068)

摘要: 阐述了一种基于 IMA 架构的航电软件分布式测试平台,该平台节点分为服务器端和客户端两类,通过互联网连接,客户端运用 Java 技术构建测试任务提交和结果获取,服务器端采用与目标机相互配合的方式实现客户端测试任务的调度和管理、测试程序的自动编译、链接、加载、执行和测试结果的存储、收集。该平台可实现有限的 IMA 硬件资源共享,提高资源利用率与测试效率,并可对 IMA 软件进行有效、充分的配置项或系统测试,为 IMA 航电软件的远程测试提供了有效的解决方案。

关键词: 航空电子; 综合模块化; 航电软件; 分布式测试; 测试代理; 测试管理

中图分类号: TP31 **文献标志码:** A **doi:**10.3969/j.issn.1671-637X.2018.02.015

Distributed Testing Platform for Integrated Modular Avionics Software

ZHOU Min-gang, MU Ming, WANG Quan, ZHANG Lei

(Aeronautics Computing Technique Research Institute, AVIC, Xi'an 710068, China)

Abstract: An avionics software distributed testing platform based on Integrated Modular Avionics (IMA) is described. There are two types of the platform nodes, Server and Client. They are connected by using the Internet. Using Java technologies, Client constructs the submission of the test assignments and the result access. Coordinating with the target machine, Server can realize the dispatching management of the test assignments, the automatic compiling, linking, loading, and executing of the test programs, and the storage and collection of the test results. Resource sharing of the limited IMA hardware can be realized on this platform. The resources utilization rate and the testing efficiency are increased, and an effective and complete test of the configuration items or the system of IMA software can be carried out. The testing platform supplies an effective solution to the long-distance testing of IMA avionics software.

Key words: avionics; integrated modularization; avionics software; distributed testing; test agent; test management

0 引言

航空电子系统是战斗机作战能力的最重要元素之一,是飞机的中枢神经系统。近几十年以来,电子技术的飞速进步、作战需求的日益复杂,推动了航空电子技术的迅速发展。航空电子系统结构经历了从分立式、联合式到综合式的变革过程^[1]。综合模块化航空电子(Integrated Modular Avionics, IMA)结构中,机载计算机被设计为通用化、模块化、开放式的公共平台,为各应

用提供服务,各 IMA 模块之间通过高速网络链接^[2]。国外已经将 IMA 结构系统成功运用于空客 380、波音 777、F-22 和 F-35 等飞机的航电系统。

伴随航空电子系统的迅速发展,软件作为航空电子系统的中枢神经系统,代码规模呈几何级数增长,软件功能实现率以 50% 的速度增长。美国第二代分立式结构战机 F-111 中,代码规模为 51 kByte,实现 20% 的航电系统功能;第三代联合式结构战机 F-16 中,代码规模为 128 kByte,实现 40% 的航电系统功能;第四代综合式战机 F-35 中,代码规模高达 500 万行,实现 80% 的航电系统功能。为解决 IMA 结构中大规模软件的测试和验证问题,国外制定了技术规范,并开发了配套的工具支持 IMA 软件的验证和测试^[3],但是这些技

收稿日期:2017-03-20 修回日期:2017-05-09

基金项目:国防基础科研项目(A0520132029)

作者简介:周敏刚(1977—),男,陕西凤翔人,学士,高工,研究方向为软件测试与验证。

术规范具体细节暂未公开。

随着我国航空技术的发展,IMA 结构已经开始应用于新一代飞机。但由于国内目前在应用的初始阶段,相关技术规范比较匮乏,仅有一些针对 IMA 软件开发方面的,对测试和验证方面暂时无相关技术要求。随着对 IMA 软件测试的实践,原有软件测试方法已经不能满足 IMA 软件的测试要求,主要体现在:应用软件已经成为供应商的独立产品,只提供应用软件的供应商一般无价格昂贵的 IMA 硬件设备,即使有硬件,也无相应蓝图配置,无法进行配置项或系统测试;IMA 软件测试不仅需要测试应用软件功能,而且需要测试运行蓝图配置、分区集成等;IMA 软件接口测试^[4]不仅包括高速网络等外部硬件接口测试,而且包括队列端口、消息端口、采样端口等外部软件接口测试。

为了解决上述问题,本文提出一种基于网络的分布式多用户 IMA 软件测试平台体系架构,并在某飞机 IMA 软件测试中进行了验证,为综合式航空电子系统中的 IMA 软件测试提供有效手段。

1 IMA 软件结构和测试方法

由于 IMA 软件在通用平台上实现信息综合与处理,构成软件密集型系统,故基于专用硬件环境的传统软件开发方法不再适用于 IMA 软件开发^[3]。为了实现 IMA 产品的通用化和模块化,IMA 软件系统采用开放式软件体系结构^[5]。在实现技术上,IMA 系统根据软件的不同用途,将软件进行逻辑分层,各层之间逻辑隔离,只能依靠特定接口进行通信^[6],这种分层策略实现了应用软件、操作系统与底层硬件的逻辑隔离,使得应用软件、操作系统和硬件设备的升级变化相互独立,互不影响。

目前国内航空领域使用较多的 IMA 软件体系结构为 ARINC653 结构,此结构将软件从底层到顶层分为模块支持层、操作系统层和应用层 3 层^[7],图 1 为其结构示意图。模块支持层主要实现对硬件设备和资源的驱动。操作系统层可细分为核心操作系统层和分区操作系统层,核心操作系统层完成分区操作系统管理、健康管理和其他管理。分区操作系统层驻留在本分区内,实现本分区进程管理、进程调度和资源管理。应用层软件是指为实现航电系统功能而开发的全部应用软件。

按图 1 中的 3 层结构,IMA 软件测试可分为操作系统测试、单分区测试、多分区测试和系统测试 4 个阶段。操作系统测试确认核心操作系统和分区操作系统功能、性能满足要求,并且接口符合 ARINC653 标准,并

保证操作系统能正确访问所有硬件设备^[8];单分区测试的目的是发现分区内软件存在的缺陷,保证分区内软件能正确实现预期的功能、性能要求,并且与其他分区通信接口正确;多分区测试是采用某种集成策略,增量进行分区集成测试,目的是发现分区间通信错误及蓝图配置是否满足各分区运行要求^[9];系统测试是将航电系统中单个 IMA 系统与其他子系统集成为完整的系统,在 IMA 硬件平台中进行确认测试,主要测试各子系统之间的通信是否满足系统定义并且正确。

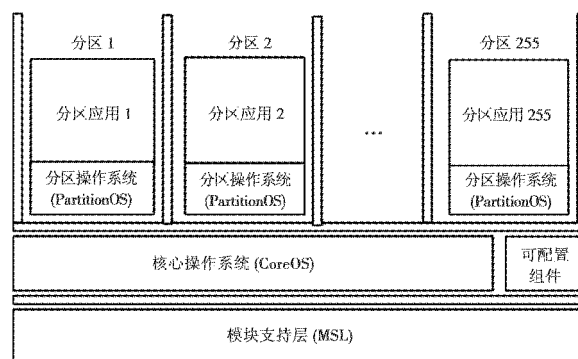


图 1 ARINC653 的软件结构示意图

Fig. 1 Software structure based on ARINC653

2 IMA 软件分布式测试平台概述

2.1 IMA 软件分布式测试平台需求分析

如果 IMA 软件使用传统软件测试方法进行测试,存在下列问题:1) IMA 硬件及其资源有限,由于传统软件测试环境为真实目标机,且测试过程中目标机通常由开发方提供,IMA 结构将许多联合式结构的设备供应商转变为应用软件供应商,其提供的应用软件只运行在 IMA 系统的某个分区中,由于 IMA 硬件平台昂贵,许多软件供应商自己也面临无目标机调试软件的困境,更无法提供给测试使用;2) 传统测试环境使用效率较低,传统软件测试过程中,测试人员占用测试环境后,即使自己在编写测试程序时,其他人也无法使用目标机,导致目标机资源浪费,并造成测试效率低下,在 IMA 系统中,几十个软件供应商如继续使用这种方式测试,不仅会造成资源的极大浪费,而且会严重影响研制进度。

为了解决 IMA 软件测试过程中目标机资源匮乏的问题,可基于互联网搭建 IMA 分布式测试环境,完成对 IMA 软件的远程测试^[10],不仅提高稀缺资源利用率,而且提高了测试效率,保证了研制进度^[11]。

2.2 IMA 软件分布式测试平台体系架构

分布式测试平台体系架构如图 2 所示。

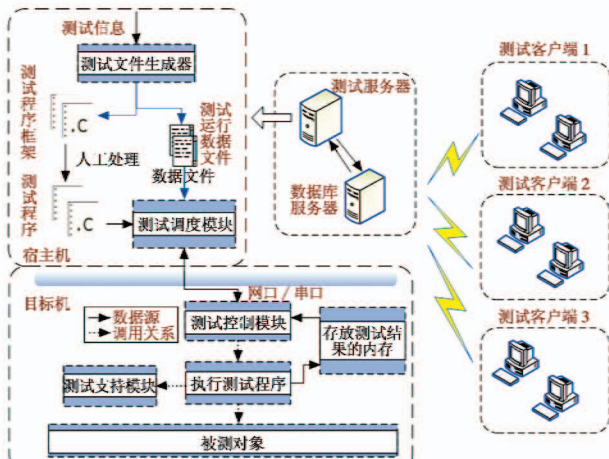


图2 IMA软件分布式测试平台体系架构

Fig. 2 System structure of distributed testing platform for IMA software

分布式测试平台包括3部分,即服务器中的测试调度、目标机中的测试控制和测试支持、客户端的数据文件生成。服务器、客户端之间通过互联网连接,服务器和目标机之间可通过互联网或其他接口(比如RS232串口等)连接。分布式测试平台工作过程如下所述。

1) 客户端的数据文件生成功能首先使分布在不同地区的测试人员产生测试程序框架和测试运行数据文件;然后可利用工具辅助生成相应测试要求的测试序列,根据测试序列提示,在测试程序框架中添加测试用例,形成完整的测试程序;最后通过互联网将测试程序和测试运行数据文件上传至测试服务器。

2) 测试服务器中运行的测试调度检测到上传的测试文件后,首先分析运行数据文件,查找运行数据相同的测试程序,标记该测试运行数据文件;然后分析map文件,整理标识映射的分配地址,产生标识与地址的映射表;然后将测试程序列表中的测试程序编译、链接为可执行代码。

3) 测试调度通过服务器与目标机接口(网口或串口)将目标码加载到目标机,驻留目标机的测试控制器负责按照测试运行数据文件控制测试程序运行,运行过程中测试程序需要调用测试支持函数实现记录实际运行结果、比较预期结果与实际结果、桩函数处理等,测试支持函数集成在测试控制器中。测试过程中,测试控制器与测试调度之间通信将测试日志文件回送到测试服务器。

4) 测试程序运行完成后,测试调度可以对测试程序、测试日志进行分析、处理,生成测试报告。

2.3 IMA软件分布式测试平台功能组成

分布式测试平台由测试调度模块、测试控制模块和测试支持模块组成,其组成见图3。

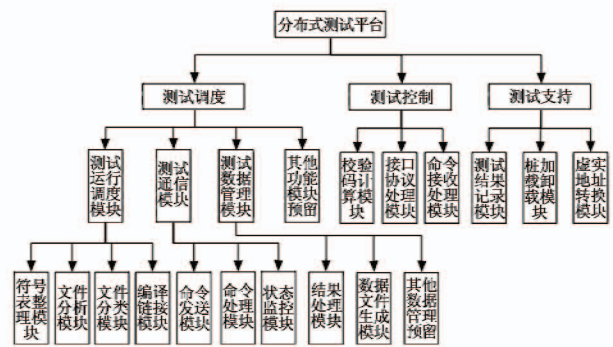


图3 分布式测试平台功能组成

Fig. 3 Composition of distributed testing platform

测试调度模块包括测试运行调度、测试通信、测试数据管理和其他功能(预留)4个子模块。测试调度模块可选用便于进行功能扩展的开发环境。

3 IMA软件分布式测试平台设计

测试调度运行在测试服务器(即与目标机相连的机器)端。服务器处理远程提交任务的调度算法如下。

```
While CurTaskFile is not NULL
  CurIP = getIPof(CurTaskFile)
  For (i = 1; i <= Numberof(CurTaskFile)/2; i++)
    If IsMatch(CFile, XMLFile) = true
      If IsLegal(XMLFile) = true
        TestQueue.add(CurTaskFile)
      ElseIf
        SetError(XMLFileIsIllegal)
      ErrorQueue.add(CurTaskFile)
    ElseIf
      SetError(XMLFileIsNotMatch)
      ErrorQueue.add(CurTaskFile)
    End if
  End For
  CurTaskFile = getNextTaskFile()
End while
```

测试等待队列的调度算法如下。

```
While WaitQueue.IsNotEmpty = true
  CurXMLFile = WaitQueue.getCurNode()
  While ReadFile(CurXMLFile) != EOF
    getCurPath(CurXMLFile)
    nd = getRoot(CurXMLFile)
    While nd is not NULL
      getTokenName(CurXMLFile)
      getMainFunc(CurXMLFile)
    End while
  End while
End while
If Tagof(os) = true and Tagof(pos) = true
  If StatsOfProgram() == CoreLayer
    SetTypeOfProgram(CoreLayer)
    CreateTestDriver()
    CopyTestProgramToOS()
    os = new Os(); pos = new Pos();
```

```

Elseif
If StatsOfProgram() == PartitionLayer
SetTypeOfProgram(PartitionLayer)
CreateTestDriver()
CopyTestProgramToOS()
os = new Os(); pos = new Pos();
p1 = new Project(); p2 = new Project()
End If
End If
GetAddrList(); CreateProcessFileForLoad()
If LoadTestProgram() = False
ReloadTestProgram()
EndIf
WaitQueue.Dequeue();
UpdateQueuePointer()
EndIf
End While

```

测试运行调度的运行步骤为:

1) 扫描服务器中是否有新的测试程序,如果有,则对新的测试运行数据文件内容进行分析,并按照测试程序运行数据进行分类;

2) 按分类对测试程序进行编译、链接,生成可执行代码;

3) 根据软件编译时产生的 map 文件,生成标识和地址映射表;

4) 准备好可执行代码与标识地址映射表后,向测试通信模块发出准备好信息。

测试通信的运行步骤为:

1) 接收到测试运行调度发送的准备好信息后,通过服务器与目标机接口向测试控制器发送加载命令;

2) 收到测试控制器回送的握手信息后,将目标码和映射表发送给测试控制器进行加载;

3) 收到测试控制器发送的加载成功信息后,发送执行命令;

4) 收到测试控制器发送执行完成信息后,发送传送测试日志命令;

5) 收到测试控制器发送的传送测试日志握手信息后,开始接收测试日志;

6) 将测试日志存入测试服务器相应位置;

7) 向测试运行调度发送执行完毕信息。

测试数据管理既可运行在测试服务器端,也可运行在测试客户端。主要完成测试程序框架和测试运行数据文件生成,以及测试报告生成等功能。测试程序框架和测试运行数据文件生成功能根据界面输入的测试信息,自动生成测试程序框架以及相应的测试运行数据文件。测试报告生成功能可根据用户的要求,从测试程序和测试日志产生相应的测试报告,并在报告中可根据测试用例和测试日志的分析结果,对软件质

量进行简单评价,而且提供开放的功能接口,方便用户根据需要进行扩展。

测试控制模块运行在目标机端,完成与测试服务器的通信、信息的解析和信息的处理,是分布式测试平台的核心功能。

测试支持模块运行在目标机端,辅助完成测试程序中测试日志的产生,以及为了模拟被测软件的故障而使用的桩函数处理等功能。测试日志和桩函数操作均在测试程序中处理,并且桩函数采用了动态加载和卸载的方法,保证了这些操作不会对被测应用软件功能和性能产生影响。

IMA 软件分布式测试平台,既结合了 IMA 软件的特点,又满足了测试规范的目标和要求。现已在符合 ARINC653 标准的国产某嵌入式实时操作系统中完成原型实现与验证工作。

4 IMA 软件分布式测试平台验证

分布式测试平台在某飞机机电管理计算机起落架分区应用软件和飞行管理软件测试中进行了应用,首轮测试效率提高 40%,回归测试效率提高 80%;在测试置信度方面,传统方法的测试结果通过串口输出,无规范格式,需要手工整理,不能快速复现,无法自动判断与统计;而采用该平台进行测试,测试日志文件由工具自动生成,格式规范,并且自动判断结果,且有统计数据。测试规范性方面,传统方法的测试程序带有测试人员个人色彩,注释取决于测试人员编程习惯,而在该平台中可自动生成测试程序框架,格式规范,注释清晰,取得了较好的结果。使用此平台与传统方法比较结果见表 1。

表 1 工具与传统方法比较

Table 1 Comparison between the tool and traditional method

比较项目	传统方法	使用此平台方法	提高效率/%
首轮效率	257 个(起落架分区 104 个,飞管 153 个)测试程序,每个测试程序从编写到调试完成平均需要 2 h,共需要 514 h	257 个测试程序,每个测试程序从编写到调试完成平均需要 1.2 h,共需要 308 h	40
回归测试效率	257 个测试程序手动运行 1 个需要 5 min,共需要 21.4 h	257 个测试程序自动运行 1 个需要 1 min,共需要 4.3 h	80

同时通过应用实践,表明该平台具有如下意义:1) 提高了 IMA 软件的测试效率,测试人员只需关注按照要求编写测试程序即可,不考虑测试程序的加载、运行等问题;2) 节约了 IMA 软件的研发成本,测试人员不需要出差进行目标机环境测试,同时,分布式布局也解决了软件开发单位硬件资源不足的问题;3) 测试平台中的桩模块 API 解决了软件测试过程中硬件故障注入

和软件破坏性故障注入问题,提高了测试覆盖率,保证了 IMA 软件测试质量。

5 结束语

随着我国新一代飞机的研制,IMA 软件在机载软件中占据的比例愈来愈大,IMA 软件测试的重要性也更加凸现,分布式测试平台可为 IMA 软件测试提供有力保障,IMA 软件分布式测试平台必将成为未来新一代综合式航电系统软件测试的重要工具。由于该系统目前支持满足 ARINC653 标准的操作系统有限,支持的 IMA 硬件平台也有限,为了提高该平台的通用性,需要在平台支持 ARINC653 标准操作系统和硬件多样化以及平台可配置化方面进行进一步研究。

参考文献

- [1] 吴志军,杨胜学. 航空电子系统发展及展望[J]. 科技资讯,2014(3):33-34.
- [2] 李林剑. 综合模块化航空电子系统[J]. 科技视界,2016,13(2):131-132.
- [3] 周庆,刘斌,余正伟,等. 综合模块化航电软件仿真测试环境研究[J]. 航空学报,2012,33(4):722-733.
- [4] 徐晓光,叶宏. 分区间通信在航空电子系统中的设计与实现[J]. 航空计算技术,2005,35(1):45-47.
- [5] 褚文奎,张凤鸣,樊晓光,等. 综合模块化航空电子系统软件体系结构综述[J]. 航空学报,2009,30(10):1912-1917.
- [6] 中国人民解放军总装备部. GJB5357-2005 航空电子应用软件接口要求[S]. 北京:总装备部军标出版发行部,2005.
- [7] 丁洁. 航空电子综合模块化操作系统体系设计分析与研究[J]. 电子设计工程,2014,22(21):169-171.
- [8] 张明杰,雷讯,杨云,等. 嵌入式实时操作系统测试理论和方法[M]. 北京:航空工业出版社,2014.
- [9] WATKINS J. 实用软件测试过程[M]. 2版. 贺红卫,杨芳,译. 北京:机械工业出版社,2004.
- [10] 崔明明. 综合模块化航空电子系统远程加载技术研究及性能分析[J]. 航空电子技术,2016,47(4):19-22.
- [11] 郑军,刘畅,任占勇. 综合模块化航空电子软件测试环境[J]. 计算机工程与设计,2011,32(8):2737-2740.
- (上接第 37 页)
- moving object detection using continuing tracking optical flow[J]. Journal of Chinese Computer Systems, 2014, 35(3):642-647.
- [2] LI T C, VILLARRUBIA G, SUN S D, et al. Resampling methods for particle filtering: identical distribution, a new method, and comparable study[J]. Frontiers of Information Technology and Electronic Engineering, 2015, 16(11):969-984.
- [3] AVIDAN S. Ensemble tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(2):261-27.
- [4] LIU B Y, HUANG J Z, YANG L, et al. Robust tracking using local sparse appearance model and k -selection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence (S0162-8828), 2013, 35(12):2968-2981.
- [5] YANG F, LU H, YANG M H. Robust superpixel tracking [C]//IEEE Transactions on Image Processing, 2014:1057-7149.
- [6] 莫邵文,邓新蒲,王帅,等. 基于改进视觉背景提取的运动目标检测算法[J]. 光学学报,2016,36(6):204-213.
- [7] WANG S, LU H C, YANG F, et al. Superpixel tracking [C]//International Conference on Computer Vision; IC-CV, 2011, 24(4):1323-1330.
- [8] ZHOU X, LI X, HU W M. Learning a superpixel-driven speed function for level set tracking[C]//IEEE Transactions on Cybernetics; TCYB, 2015, 46(7):2168-2267.
- [9] LIU J J, CHEN Y, ZHA C, et al. Tracking using superpixel features[C]//Eighth International Conference on Measuring Technology and Mechatronics Automation; ICMTMA, 2016, 211:878-881.
- [10] HUANG G H, PUN C M, LIN C. Video object tracking using interactive segmentation and superpixel based Gaussian kernel [C]//The 19th International Conference on Information Visualization; iV, doi:10.1109/iv. 2015. 81.
- [11] GAO J, LING H B, HU W M, et al. Transfer learning based visual tracking with Gaussian processes regression [C]//Proceedings of ECCV, Zürich, Switzerland, 2014:188-203.
- [12] LI X, HAN Z, WANG L, et al. Visual tracking via random walks on graph model[J]. IEEE Transactions on Cybernet, 2016, 46(9):2144-2155.
- [13] NAM H, HAN B. Learning multi-domain convolutional neural networks for visual tracking[C]//Proceedings of CVPR, Las Vegas, NV, USA, 2016:4293-4302.
- [14] 刘龙,樊波阳,刘金星,等. 面向运动目标检测的粒子滤波视觉注意力模型[J]. 电子学报,2016,44(9):2235-2241.