

一种基于 XML 的任务驱动型数据传输与同步方法

乔得吉 肖卫东

(中航工业信息技术中心客户服务业务部 北京 100028)

摘 要 出于信息安全和业务需要,人们在彼此逻辑或物理隔离的局域网环境中浏览和处理信息,但是又不想丧失网络互联带来的便利和强大的信息处理能力。通过数据传输与同步机制,模拟广域网环境、避免信息孤岛是一种现实可行的选择。给出了一种基于 XML 的任务驱动的数据传输与同步方法,通过分析数据传输场景,设定了基于 XML 的任务数据模板和任务结果数据模板,借助于具体的信息化系统,基于相应的模板,自动生成任务数据包及任务数据结果包,以数据光盘进行信息传输,通过数据寻址、定位及导入操作实现了数据同步功能。通过该数据通路,实现了各级彼此独立的信息系统的互联,用户犹如在一个互联的广域网环境中进行数据处理。该方法大大提升了信息系统的安全性,降低了铺设或租用线路的成本,解决了各级网络之间不连通导致的信息壁垒,更符合现阶段特殊行业使用网络环境的要求。

关键词 局域网,数据传输与同步,XML,任务驱动,数据模板

中图法分类号 TP274 文献标识码 A

Task Driven Data Transmission and Synchronization Method Based on XML

QIAO De-ji XIAO Wei-dong

(Customer Service Business Department, AVIC Information Technology Corporation, LTD, Beijing 100028, China)

Abstract Considering the situation of information security and business requirement, people browse and deal information on local area network which is logical isolated or physical isolated with each other, but people don't want to lose the powerful data handling ability. Through data transmission and synchronization, to simulate wide area network and avoid information silo is a feasibility choice. This paper provided a task driven data transmission and synchronization method based on XML. This method analyzes data transmission scene, sets the data template of task and task result and recurs to specific information system to build task data package and task result package automatically. It transfers data by data disk, and synchronizes data by data addressing, location and import. By this data passageway, the isolated information systems are interconnected, and people deal with information just like on wide area network. This method enhances the security level of information system, reduces the cost of network line, solves the information silo caused by disconnection, and satisfies the requirements to use network of specific trade.

Keywords Local area network, Data transmission and synchronization, XML, Task driven, Data template

1 引言

计算机网络是计算机技术和数据通信技术的产物。根据传输距离的差异,计算机网络可以分为局域网、城域网和广域网 3 种。局域网(Local Area Network, LAN)是将分散在有限地理范围内的多台计算机通过传输媒体连接起来的通信网络,通过功能完善的网络软件实现计算机之间的相互通信和资源共享;广域网(Wide Area Network, WAN)是在传输距离较长的前提下所发展的相关技术的集合,用于将大区域范围内的各种计算机设备和通信设备互联在一起,组成一个资源共享的通信网络^[1]。

随着信息技术的发展和信息化产品的大规模使用,人们通过有线或无线接入的方式,获得了强大的信息处理能力,已经习惯于在一个互联的网络空间中浏览和处理各类信息。由

于广域网固有的开放性和包容性,人们在享受网络带来便利的同时,也将大量的信息资源置于危险之中。基于安全和业务需要,人们通过逻辑隔离或物理隔离的方式将预期采用和建设的广域网割裂成了一个个彼此独立的局域网,不可避免地出现了信息孤岛,可是人们又不想失去互联带来的强大的信息处理能力。这就迫切需要一种数据传输和同步机制来模拟一个互联的广域网环境,从而充分调度和利用计算机网络的潜能。

目前在数据交换协议方面的工作主要包括互联网通用协议、互联网轻量级协议和物联网轻量级协议 3 类。首先在互联网中广泛采用的协议有 XML、JSON 等,尤其以 XML 占主导地位^[2]。可扩展标记语言 XML(eXtensible Markup Language)的出现,使以网络应用为代表的一大批新型应用得以迅速发展。在当前各种网络应用中,如电子商务、电子政务、

乔得吉(1983—),男,硕士,工程师,主要研究方向为数据库技术与计算机软件, E-mail: qiaodeji@163.com; 肖卫东(1969—),男,硕士,工程师,主要研究方向为软件工程与信息安全。

金融、出版、科学数据和各种资源的数字化等,XML 扮演着极其重要的角色,它已经成为数据交换事实上的标准、SOA 架构的基石^[3]。XML 能够独立于平台表示和交换数据,应用越来越广泛^[4]。在信息化平台呈现多元化的今天,XML 语言的优势尽显,得到了越来越多的关注和应用。

本文给出了一种基于 XML 的任务驱动的数据传输与同步方法,以既定的任务进行牵引,通过事先约定和设置基于 XML 的任务模板和任务结果模板,借助于具体的信息系统,在导出并下发任务时,信息系统基于任务模板生成任务数据 XML 文件并进行打包,同时通过数据光盘进行数据传递;任务接收方接收任务数据包并导入信息系统,任务处理完成后,信息系统基于任务结果模板生成任务结果数据 XML 文件并进行打包,同时上报任务结果数据包。信息系统通过数据寻址与自动定位实现任务结果数据的导入与同步,并在信息系统中进行展示,最终在彼此隔离的 LAN 环境中实现了类似 WAN 的应用效果。

2 基于任务的数据交互

有某行业的三级机构,分别为一级机构、二级机构、三级机构,一级机构和二级机构为管理部门,三级机构为执行部门,它们分布在异地,各机构有内部的 LAN,但是彼此之间网络不连通。出于业务和工作的需要,一级机构需要经常性下达各类任务,二级机构接收任务以后,需要及时进行签批和分派,三级机构完成了具体的工作任务以后,要及时上报完成情况及结果信息,便于上级机构知悉及管控。

在任务数据传递时,按照既定的通路以数据光盘的方式进行流转,不允许在各机构之间跨级传递。基于任务的数据交互流程为:1)在信息系统(一级机构版)填写任务信息,执行导出操作,生成任务数据包并进行下发;2)传递任务数据包并导入信息系统(二级机构版),进行必要的信息项浏览及签批后,生成任务数据包并继续下发;3)传递任务数据包并导入信息系统(三级机构版);4)三级机构工作人员基于任务内容,组织开展并完成相应的工作,任务完成后,生成任务结果数据包并进行上报;5)传递任务结果数据包并导入信息系统(二级机构版),二级机构工作人员基于该任务的完成情况,签署相应的点评意见并进行上报;6)传递并导入信息系统(一级机构版),一级机构工作人员主要是浏览任务的完成情况。通过以上流程,最终完成任务数据下发和任务结果数据上报的闭环控制。数据交互场景如图 1 所示。

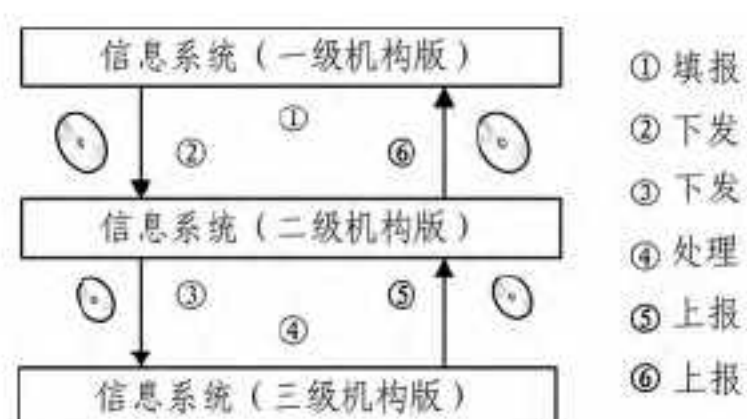


图 1 数据交互场景

3 数据交互模板设置

数据同步是将本平台需要的数据从其他系统中同步过来,以实现自身结构的组建^[5]。异构系统之间数据交换的办法之一是采用统一的信息描述格式。XML 具有自描述性和

可扩展性等优势,便于表达各种类型的数据,可作为异构系统之间的中间语言,实现不同数据源之间的信息交换和共享^[6]。XML 文档可包括多层嵌套的数据结构,因此,XML 解析通常变得比较复杂。在缺乏高效 XML 解析方法的情况下,XML 解析成为 XML 应用的性能瓶颈^[7]。XML 文档可以分为良构的文档和有效的文档。良构的文档是指符合 XML 语法规则但不一定符合某个模式(DTD 或 XML schema 等)约束的文档;有效的文档是指既符合 XML 语法规则又符合某个模式约束的文档^[8]。

在数据导入及同步时,经常遇到因数据项不统一和格式不规范而导致的同步失败问题。为了避免此类问题,事先制定相应的数据交互模板是一种通用的可行办法。为了达到更好的应用效果和规范化水平,从技术角度考虑,需要制定相应的数据交互模板嵌入信息系统,并实现与具体功能模块的调用接口;从管理角度考虑,需要制定数据同步的标准规范,并下发至各应用部门,提高其在业务工作中的数据收集和填报的标准化水平。为了提高通用化水平,确保其平台无关性,最终决定使用基于 XML 的数据交互模板。

3.1 基于 XML 的任务模板

基于 XML 的任务模板中主要包含两部分内容,分别为 XML 标记和特殊的格式约定。在 XML 标记中包含 XML 声明部分、根节点元素、枝节点元素、叶节点元素,通过父子关系将各节点元素紧紧相连,以树型结构进行展示;在特殊的格式约定部分,主要是对日期、时间等格式进行规范化约定,如时间统一使用 YYYY-MM-DD hh:mm:ss 格式。任务数据 XML 模板的形式化结构如图 2 所示。

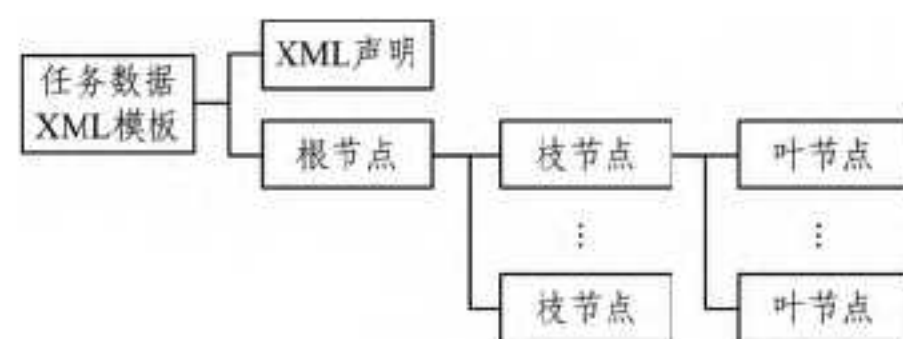


图 2 任务数据 XML 模板的形式化结构

3.2 基于 XML 的任务结果模板

与任务模板类似,基于 XML 的任务结果模板中也包含 XML 标记和特殊的格式约定两部分内容。由于本系统允许以添加导入时间戳的方式将多个任务结果包的内容信息关联到任务信息中,任务信息和任务结果信息之间呈现 1:n 的关系。XML 标记中除了制定必要的任务结果元素节点外,必须明确任务结果信息和任务信息的关联关系,本文通过强制不唯一的任务 id 节点进行关联。具体地,利用 java.util.UUID 的 randomUUID 方法,生成随机数的方式,并通过重复性校验机制,确保各任务信息中任务 id 的唯一性。该关联关系,可以方便后续的寻址与定位操作,更容易地将任务执行完毕后的结果信息导入信息系统各机构版本中。

4 任务管理模块的设计与实现

在任务管理模块,除了常规的添加、编辑、删除、查询操作外,为了方便数据的传输和同步,必须提供任务导出功能和任务结果导入功能,它们均借助特定的接口调用相应的数据模板生成相应的数据包。任务管理模块在设计时采用一体化开发、分类实施的思想,统一设计与开发后台的数据源和前台界

面。在具体实施过程中,根据组织机构的不同,将其划分为3个版本,分别为一级机构版、二级机构版、三级机构版。基于业务需要及职能划分,将界面元素进行了重新编排和裁剪。如一级机构版只能进行任务数据的导出及任务结果数据的导入,不能进行任务数据的导入及任务结果数据的导出。三级机构版本只能进行任务数据的导入及任务结果数据的导出,不能进行任务数据的导出及任务结果数据的导入。各版本的信息系统由于使用了统一的数据源,有着相同的数据结构及相同的数据操作方法,很容易实现数据的传输和同步,大大降低了出错的概率。

4.1 内置数据模板

在信息系统的开发中采用了经典的轻量级架构 SSH (Struts+Spring+Hibernate),它能够满足应用需求。其既加快了开发速度,也让信息系统显得更加轻巧,在将数据模板进行内置的过程中,将其抽象为 java 文件,该文件中明确了各数据元素及数据类型,方便导出接口和导入接口的快速调用。

4.2 数据导出接口

在数据转换及结构信息映射时,关系模式中的基本结构信息以及用户自定义约束主要保存在关系模式向量模型中,完成映射之后则保存在 XML 模式树的叶子节点中^[9]。在任务管理模块内置任务导出接口,该接口搭建了存储在数据库表中的任务记录和任务信息 XML 文件之间的桥梁。通过该接口,获取任务信息及相应的处理信息,参照已定义的数据导出模板,即可生成基于 XML 的数据文件。

4.3 数据导入接口

在任务管理模块内置任务导入接口,该接口提供了任务结果 XML 文件和存储在数据库表中的任务结果记录之间的通路。通过该接口,选择将要导入的数据文件包,即可解包并进行 XML 文件属性比对,通过自动的数据寻址与定位,将数据文件逐行导入系统。

4.4 任务进度图形显示

相较于文字和表格,图形更加直观,表现力更强。为了更直观地展示该任务驱动的数据流向,根据选中的任务信息,以路线图的方式动态展示当前数据的流向和任务的进展情况,浅色的表示已完成,深色的表示未完成,并将页面信息进行了整合,力图在有限的空间内展示更多信息。任务进度显示效果如图3所示。



图3 任务进度显示图

5 基于 XML 的数据导出

5.1 基于模板的任务数据导出及打包

在任务数据导出时,信息系统首先调用数据导出接口,打开数据导出模板 java 文件,读取模板文件中的属性和特定的

格式规范,并基于将要导出的任务信息,对模板中的属性逐一进行赋值,最终形成符合数据模板规范要求的 XML 数据文件。

为了减少数据传输量和文件数量,在互联网中通常采用的方式是先压缩再传输^[2]。在文件生成完毕之后,将该 XML 文件连同相应的附件信息进行压缩打包,并且按照既定的命名规范对该压缩包进行重命名后存储在相应的磁盘位置,同时更新任务状态(如将状态由“一级机构填报”变更为“一级机构分派”)。基于模板的任务数据导出及打包流程如图4所示。

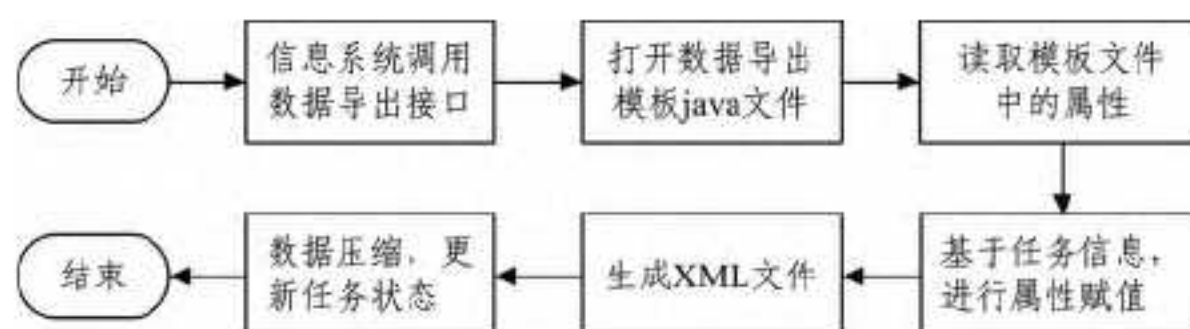


图4 基于模板的任务数据导出及打包流程图

6 基于 XML 的数据定位及导入

任务数据包沿着既定路线在各机构之间进行传递,三级机构接收到任务数据包之后,解压并导入信息系统(三级机构版),浏览任务信息的具体内容,依据任务要求完成相应的处理工作。工作完成后,生成基于 XML 的任务结果数据,并进行上报,上报数据在导入信息系统时自动进行数据寻址与定位,并最终完成数据的导入与同步工作,数据同步完成后,通过信息系统可直观地展示和查看结果信息。任务结果数据导出、定位及导入流程如图5所示。

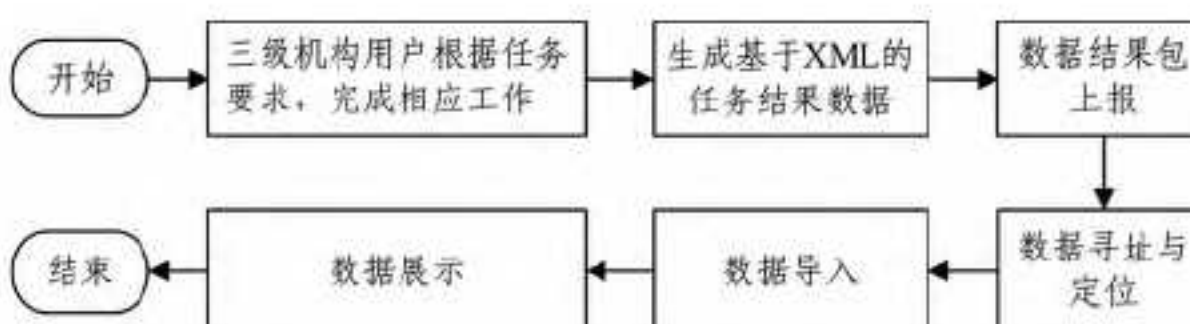


图5 任务结果数据导出、定位及导入流程图

6.1 基于模板的任务结果数据导出及打包

与基于模板的任务数据导出及打包类似,基于模板的任务结果数据导出及打包也是调用数据导出接口,根据任务结果数据模板和任务结果数据生成 XML 文件并进行压缩,并且按照既定的命名规范对该压缩包进行重命名,也将压缩后的数据包存储在相应的磁盘位置,以便于刻录光盘,同时更新任务状态。

6.2 数据寻址与定位

获取任务结果数据包后,在执行导入操作时,需要先进行数据的寻址与定位。只有寻址成功后,才执行数据导入操作,否则导入工作没有意义。数据寻址与定位的步骤为:1)解压任务结果数据包,判断是否存在数据文件,并判断文件格式是否为 XML;2)获取任务结果 XML 文件,逐行判断数据结果集中的元素节点是否符合规范要求,然后判断元素文本是否符合格式要求;3)在关系数据库中查询任务记录表;4)找到任务记录表后,根据任务结果 XML 文件中的任务 id,查询任务记录表中该行的记录;5)在数据库中查询任务结果记录表;6)在任务结果记录表中进行定位,等待数据插入。数据寻址与

定位原理如图 6 所示。

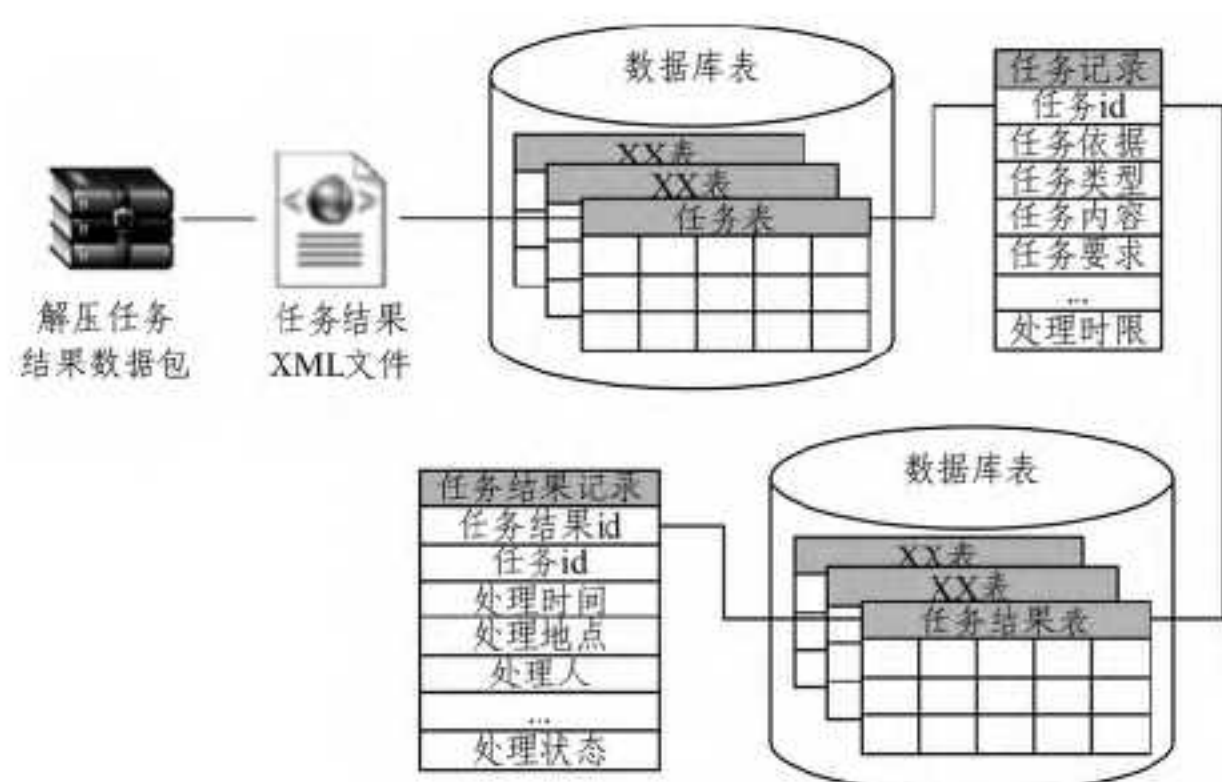


图 6 数据寻址与定位原理图

6.3 数据导入与同步

在数据导入的过程中,必须考虑版本的问题,同一个数据结果包有可能被多次导入,数据包中的内容可能不尽相同,但数据结果包只要符合既定的格式规范,就应该被成功导入,但是不允许进行覆盖操作,必须保留每一次导入的详细信息,即允许符合数据规范的数据结果包被多次重复导入,只是需要在每次导入时添加导入时间戳以示区分。

处理数据同步的过程就是对多个局部数据库同时进行添加、修改、删除操作,以实现异构数据库的数据同步^[10]。XML 序列化是将文件转换成字符串的过程,其目的有两点:一是压缩文件,减少冗余数据;二是预处理,方便以字符流进行操作。序列化的主要思想是用元素标签(Tag)来表达 XML 数据,包括 XML 文件中的元素(Element)、属性(Attribute)、文本(Text)^[2]。本文在具体执行数据导入时,首先定位待导入的 XML 文件,从 XML 文件中读取结果集并将其转化为数据流,并存储在列表 List 中;同时通过对象拷贝操作,将任务结果信息拷贝到任务结果记录表中,并更新任务记录的任务状态信息(如将任务状态信息由“二级单位上报”改为“一级单位接收”)。通过任务 id 和任务表建立主子表关联关系,从而实现了数据的同步工作。

6.4 数据结果展示

数据导入最终的目的在于数据的同步及结果展示。在数

据导入成功后,通过数据结果展示页面直观地显示任务数据的流转及执行情况。由于允许数据包被多次重复导入,在任务数据结果展示时,默认只展示最后一次导入的信息。同时系统提供了浏览历史版本信息的功能,通过导入时间戳信息,即可浏览该时间戳对应的导入信息。

结束语 本文以任务为驱动,基于 XML 语言和数据交互场景,给出了一种数据传输与同步方法,通过制定数据模板,避免了数据识别和出错问题,通过数据导出及导入接口,数据光盘传递,使得工作在彼此隔离的局域网环境中的信息系统实现了犹如在互联的广域网环境中的效果,解决了跨地域的信息沟通交流问题。实践证明,该方法能更好地适应特定的行业现状,在网络不连通的环境下,也可进行业务数据的传输和处理,既提高了信息系统的安全保密水平,也降低了对通信线路的连通要求。

参 考 文 献

- [1] 希赛教育软考学院. 系统架构设计师教程(第3版)[M]. 北京: 电子工业出版社, 2014
- [2] 侯陈达, 李栋, 邱杰凡, 等. EasiDEF: 一种水平化轻量级物联网数据交换协议[J]. 计算机学报, 2015, 38(3): 602-613
- [3] 周军锋, 孟小峰. XML 关键字查询处理研究[J]. 计算机学报, 2012, 35(12): 2459-2478
- [4] 陈华城, 杜学绘, 陈性元, 等. 基于 D-S 证据理论的 XML 文档潜在信息获取算法[J]. 计算机应用研究, 2013, 30(4): 1187-1190
- [5] 郑扬飞, 金辉, 张勇, 等. 消防一体化环境下的信息交换平台关键技术研究[J]. 计算机工程与应用, 2012, 48(7): 219-223
- [6] 曹兰英, 严义, 郭惠峰. 基于模式匹配的 XML 自动转换技术[J]. 计算机工程与应用, 2012, 48(25): 72-76
- [7] 方跃坚, 余枝强, 翟磊, 等. 一种混合并行 XML 解析方法[J]. 软件学报, 2013, 24(6): 1196-1206
- [8] 鉴保瑞, 宋余庆, 陈健美, 等. 一种基于关系的 XML 文档模型映射方法[J]. 计算机应用研究, 2011, 28(12): 4621-4624
- [9] 侯莹, 李凤岐, 牛纪桢, 等. 关系模式到模块化的 XML Schema 的模型映射方法[J]. 计算机工程与应用, 2011, 47(12): 122-125
- [10] 申利民, 李卫东. 面向协同系统集成的数据同步模型[J]. 计算机应用研究, 2012, 29(4): 1384-1386

(上接第 466 页)

- [10] Zhu Q S. Improving Program Performance via Auto-Vectorization of Loops with Conditional Statements with GCC Compiler Setting[J]. Applied Mechanics and Materials, 2014, 433: 1410-1414
- [11] Ojha D K, Sikka G. A Study on Vectorization Methods for Multicore SIMD Architecture Provided by Compilers[C]// ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I. Springer International Publishing, 2014: 723-728
- [12] 李春江, 黄娟娟, 徐颖, 等. 典型编译器自动向量化效果评估与分析[J]. 计算机科学, 2013, 40(4): 41-46
- [13] King J. Symbolic execution and program testing[J]. Communications of the ACM, 1976, 19(7): 385-394

- [14] Bardin S, Kosmatov N, Cheynier F. Efficient Leveraging of Symbolic Execution to Advanced Coverage Criteria[C]// 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation (ICST). IEEE Computer Society, 2014: 173-182
- [15] Ibing A. SMT-Constrained Symbolic Execution for Eclipse CDT / Codan [M] // Software Engineering and Formal Methods. Springer. International Publishing. 2013: 113-124
- [16] Reus B, Charlton N, Horsfall B. Symbolic Execution Proofs for Higher Order Store Programs[J]. Journal of Automated Reasoning, 2015, 54(3): 199-284
- [17] Chen T, Zhang X S, Ji X L, et al. Test Generation for Embedded Executables via Concolic Execution in a Real Environment[J]. IEEE Transactions on Reliability, 2015, 64(1): 284-296