

# SG-APS 应用服务器中间件的设计与实现

许大卫,吴 舜,孟 德,郑晓琳

(国网冀北电力有限公司信息通信分公司 信息通信运检中心,北京 100053)

**摘 要:**应用服务器中间件在电力信息化系统中占有举足轻重的地位,研发具有自主知识产权的应用服务器是实现电力系统信息安全战略的关键。SG-APS 应用服务器中间件项目主要基于开源技术研发中间件软件,构建公司基础软件平台,提升基础软件安全自主可控能力,摆脱厂商依赖。介绍了国家电网公司 SG-APS 应用服务器的设计与实现,分析了该应用服务器在电力信息系统环境下的应用需求,描述了该应用服务器的系统架构、设计思想和关键技术。目前,SG-APS 应用服务器中间件 1.0 版本的开发工作现已基本完成,并将在国家电网信息化系统中逐步替换国外产品。

**关键词:**中间件;应用服务器;国家电网;SG-APS

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2016)10-0127-06

doi:10.3969/j.issn.1673-629X.2016.10.028

## Design and Implementation of SG-APS Application Server Middleware

XU Da-wei, WU Shun, MENG De, ZHENG Xiao-lin

(Information and Communication Center, State Grid Jibei Information & Telecommunication  
Company, Beijing 100053, China)

**Abstract:** Application server middleware possesses an extremely important position in electric power information systems. Developing application server with independent intellectual property is essential to the information security strategy of the electric power systems. The project of SG-APS application server middleware mainly bases on open source technology research and development middleware software, and builds the company's basic software platform to enhance the ability to independently control the basic software security capabilities, getting rid of manufacturers rely on. It introduces the SG-APS application server middleware developed by State Grid Corporation, and analyzes its application requirements under the environment of the electric power information systems, and describes its architecture, design, implementation and key technologies. At present, the development of the 1.0 version of SG-APS application server middleware has been basically completed, and it will gradually replace foreign products in the national grid information system.

**Key words:** middleware; application server; State Grid; SG-APS

## 0 引 言

随着我国电力行业信息化水平的不断提升,电力信息化系统的应用范围越来越广,复杂程度也越来越高。国家电网公司的大部分信息系统均采用 J2EE 标准<sup>[1]</sup>,其中应用服务器中间件作为 J2EE 标准的核心中间件,在电力信息化系统中占有举足轻重的地位<sup>[2]</sup>。以往国家电网公司通常都是采购国外商用应用服务器中间件产品,如 WebLogic 等<sup>[3]</sup>。随着电力信息化建设要求不断提高,这种方式逐渐显现出越来越多的弊端。

首先,电力行业是国民经济中具有举足轻重地位的核心领域,长期依赖国外产品无法保障“自主可控、

安全可靠”的国家信息安全战略目标。其次,随着应用部署节点的不断增加,采购与升级成本越来越高。第三,商用应用服务器安装与部署时对 IT 基础条件要求相对较高、补丁升级流程较长、售后服务响应不及时。第四,商用应用服务器虽然均基于 J2EE 标准,但各自的配置、管理方式千差万别,每个系统都需要独立采购,难以统一管理和维护<sup>[4]</sup>。

为了确保电力系统的信息安全,防范信息泄露,摆脱对国外厂商的依赖,掌握核心技术,国家电网公司亟需研发适用于国内电力信息化建设需求,拥有自主知识产权的应用服务器中间件<sup>[5]</sup>。通过应用服务器中间件的研发,提高国家电网信息化运维效率,减少运维投

收稿日期:2015-11-06

修回日期:2016-03-04

网络出版时间:2016-09-19

基金项目:中央高校基本科研业务费(JB150407);数据(灾备)中心-中间件软件——期项目(7199001402M0)

作者简介:许大卫(1984-),男,工程师,从事信息技术及管理的工作。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160919.0839.002.html>

入,降低运维难度,配合应用服务资源池的建设,实现应用按需分配和自主运维。

文中介绍了国家电网公司研发的标准、安全、可控、高可用并具有丰富功能的 SG-APS 企业级应用服务器,分析了该应用服务器在电力信息系统环境下的需求,描述了该应用服务器的系统架构、设计和关键技术。

## 1 需求分析

国家电网公司 SG-APS 应用服务器中间件在定位上主要考虑两个因素。

(1)应用服务器中间件在国际上已经形成了标准。国家电网多年来也在这些标准下建设了大量的信息系统。从延续性和兼容性上来说都要求国家电网应用服务器遵循国际标准。

(2)国家电网公司已将云计算作为未来信息化发展的方向。从云平台的角度来看,能够承载服务化的云应用是对应用服务器的根本要求。这就要求应用服务器在国家电网云平台环境下能够按照云平台的要求实现应用支持。此外,应用服务器研发还必须考虑在当前国家电网的云平台环境尚未完全建成的前提下,如何在开发中与云平台的建设保持同步,甚至适当超前于云平台建设的要求。

### 1.1 功能需求

应用服务器需要把不同业务应用作为构件整合到一个协同工作的环境里,并为应用提供名字、事务、安全、消息、数据访问等服务,包括:

(1)标准和规范。包括 Java EE 5 的相关 13 种技术规范<sup>[6]</sup>、主流的 Web 服务标准和互操作标准。

(2)容器和组件。包括 Web 容器、EJB 容器、RMI 服务容器、Web 服务容器、JCA 服务、数据库连接池、事务控制组件、缓存组件、高级消息传输、Rest 服务<sup>[7-14]</sup>等,并支持各种成熟开发框架。

(3)集群和动态扩展。实现集群部署的动态管理,支持动态加载新业务;提供跨多种平台服务器的集群部署配置以及故障切换功能,从而快速适应企业现有软硬件环境并可确保关键应用和服务高效可用。

(4)管理配置。具备标准的 B/S 模式管理控制台,可对远程的应用服务器环境进行应用部署、管理维护和监控;支持对系统运行状态、Web 应用、数据库连接池、事务服务、JMS 服务及其他服务端组件与服务进行动态监控与管理;提供应用移植工具,实现应用程序包从其他应用服务器到本应用服务器环境的移植。

### 1.2 非功能需求

(1)安全。提供基于容器的安全策略,限制对应用的访问,保障企业数据的安全,防止恶意攻击。通过

使用应用服务器提供的监控管理工具对服务的运行情况进行实时跟踪监控,并提供方便的日志管理功能以使用户进行审计。支持 JAAS 协议,支持用户采用第三方的安全策略集成企业级应用,以增强应用安全性。

(2)可靠性。应用服务器应对应用提供自我恢复的能力管控,即知道在应用是否有问题,以及支撑的 Java 容器是否出现故障时,对出现问题的应用进程、组件提供分析、恢复、重启的能力。如出现死锁或者其他原因崩溃的操作系统虚拟机(节点),应用服务器可以根据不同的定制策略,选择性重新启动恢复。

(3)性能。性能指标为:URL 访问应答时间<5 ms,瞬间最大并发访问请求>2 000,大于 1k 文件 JSP 文件编译时间>10 ms,大于 1k 文件 class 文件加载时间>10 ms,URI 路径解析时间<0.1 ms,同时需要:

- 满足系统 7×24 小时业务无间断,稳定可靠运行。
- 日常平均 CPU 占用率<40%,忙时<75%,内存占用率<50%,最大并发时<75%。
- 系统应满足易理解、易分析、易配置、易修改、易测试的要求。
- 系统从用户体验维度出发,满足页面布局合理,通用操作规范,出错处理、反馈与提示人性化等要求。
- 具备开发架构和动态扩展能力。

此外,应用服务器还要综合考虑数据库、操作系统、硬件环境、安全防护和节点冗余等问题。

## 2 设计与实现

### 2.1 系统架构

应用服务器中间件要实现 Java EE 5 规范规定的各种标准组件和标准服务,包括 Web 容器、EJB 容器、RMI 服务容器、Web 服务平台、JCA 服务、数据库连接池、事物控制组件等,并支持各种成熟开发框架,如图 1 所示。SG-APS 应用服务器中间件主要实现了以下核心组件和服务。

#### 1)安全。

基于 Java EE 标准的安全服务<sup>[15]</sup>,是动态可扩展的安全体系结构,包括基于 JMAC 的验证和基于 JACC (Java Authorization Contract for Containers,Java 容器授权合同)的授权,使得服务器能够接入第三方认证、授权模块。

消息安全性使服务器可以在消息层执行 Web 服务调用和响应的端对端验证。消息安全性提供者提供了请求和响应消息所需的验证类型等信息。支持的验证类型包括发件人验证(用户名和密码验证)和内容验证(XML 数字签名)。

#### 2)事务。

事务按照 JTA1.0.1B 规范实现,并且支持如下功能:

(1)本地事务优化。对于使用同一个资源管理器的操作,事务管理器使用本地事务机制,本地事务只存在于一个事务过程中,采用一阶段提交;只有涉及到多

个资源管理器的操作才使用全局事务,全局事务可参与多个事务过程。

(2)事务恢复功能。在服务器启动时从故障中恢复并完成被搁置的事务。

(3)事务超时控制。自动回滚超时的事务。

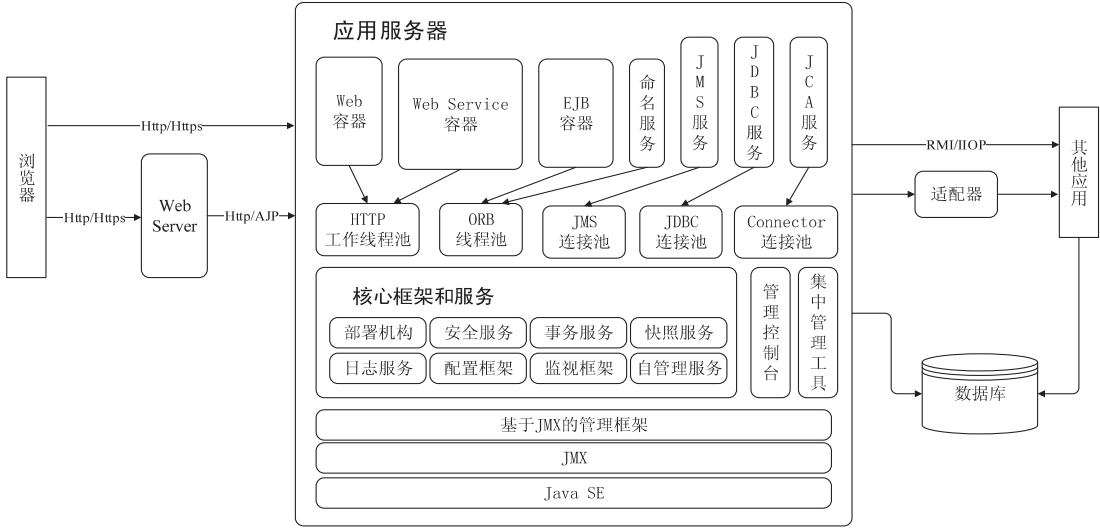


图 1 系统架构

3)日志。  
基于 Java Logging API(JSR-047)实现日志管理。

4)Web 容器。  
Web 容器提供了 Web 应用的运行时环境,包括生命周期管理、安全、请求转发等。同时 Web 容器为 Web 应用提供访问其他 API 的能力,如命名服务。

5)EJB 容器。  
EJB 容器实现了 EJB3.0 的规范,提供了 EJB 运行的环境。它负责 EJB Bean 实例的生命周期管理,Bean 实例状态管理,并发(线程)管理,事务服务,安全服务,命名服务,资源(数据源等)管理,使得 EJB 客户端程序能进行远程调用或本地调用。EJB 容器支持 EJB3.0 规范和 JPA,提供定时服务、依赖注入、拦截器、基于 EJB 的 Web Service 等功能。

6)Web Service 容器。  
Web 服务定义了应用程序如何在 Web 上实现互操作性,它使用 XML(可扩展标记语言)表示数据的基本格式,使用 DTD(Document Type Definition)或 XSD(XML Schema)定义数据类型及扩展这套数据类型,通过 XML 文件(如 Web 服务定义语言:WSDL)描述 Web 服务应用程序及其函数、参数和返回值,并通过基于 XML 的协议(如简单对象访问协议:SOAP)向客户机提供标准的 RPC 方法来调用该 Web 服务应用程序。

7)资源管理器。  
资源管理器负责管理 SG-APS 应用服务器中间件的各种资源,主要包括:

BC 连接池。JDBC 资源为应用程序提供了连接数据库的方法。要创建 JDBC 资源,需要指定与其关联的连接池。多个 JDBC 资源可以指定一个连接池。JDBC 连接池维护特定数据库的一组可重复使用的连接。应用程序请求连接时可以从池中获取一个连接。应用程序关闭连接时,连接将返回到池中。

(2)JMS 消息服务资源。有四个组成部分:JMS 服务提供者、消息管理对象、消息的生产者、消费者和消息本身。

(3)JavaMail 资源。通过调用 JavaMail API 实现从 Java 应用程序访问网络或 Internet 上支持 Internet 消息访问协议(Internet Message Access Protocol,IMAP)和简单邮件传输协议(Simple Mail Transfer Protocol,SMTP)的邮件服务器。它本身不提供邮件服务器功能,必须有权访问邮件服务器,才能使用 JavaMail。

(4)Connector 连接器资源。通过 JCA(Java Connector Architecture)提供了一个应用服务器和企业信息系统(EIS)连接的标准 Java 解决方案,以及把这些系统整合起来的方法。JCA 简化了异构系统的集成,用户只要构造一个基于 JCA 规范的连接器应用,并将该连接器应用部署到 J2EE 服务器上,这样不用编写任何代码就可以实现 EIS 与 J2EE 应用服务器的集成。

2.2 系统设计与实现

为缩短开发周期,降低开发风险,SG-APS 应用服务器中间件参考了开源应用服务器软件 JBoss Application Server。

JBoss Application Server<sup>[16]</sup>是开源社区著名的应用

(1)JDBC 数据源。主要实体为 JDBC 资源和 JD-

服务器项目,遵循 LGPL 2.1 许可协议。JBoss Application Server( 现已更名为 WildFly)以其架构清晰、高效、稳定、易用等优势,先后为众多大型应用系统所采用。

SG-APS 应用服务器中间件沿用了 JBoss Application Server 的核心容器和核心服务,重点对其中的服务器参数配置和服务器运行监控两个客户端工具进行了汉化、消缺、界面美化和易用性改造,新增了集群日志、自动运维工具接口和安装工具三个组件。

(1) 服务器参数配置客户端。

服务器参数配置客户端的核心类主要有 3 个,其中 Constants 接口以字符串常量的形式定义了请求方法( Request Methods)、协议( Protocols)、标题头( Headers)、类型( Content Types)、字符集( Charsets)、状态码( Status Codes)。

ConsoleMode 是一个枚举类,用于定义展现不同的管理控制台的操作模式,如 CONSOLE( 展示正常的控制台);SLAVE\_HC( 如果企图访问控制台,提示错误主机是从属的);ADMIN\_ONLY( 试图访问控制台,显示服务器/主机处于管理模式错误);NO\_CONSOLE( 试图访问控制台,显示 404 错误码),同时生成对应的资源页面文件句柄。

资源句柄类 ResourceHandler 实现了 ManagementHttpHandler 接口。它通过调用 ModuleIdentifier()方法获得 Modules( 是指 JBoos7.1 的所有功能模块)的名称和 ID,提供给界面进行展示和管理。

用户管理员登录进入服务器参数配置界面如图 2 所示。

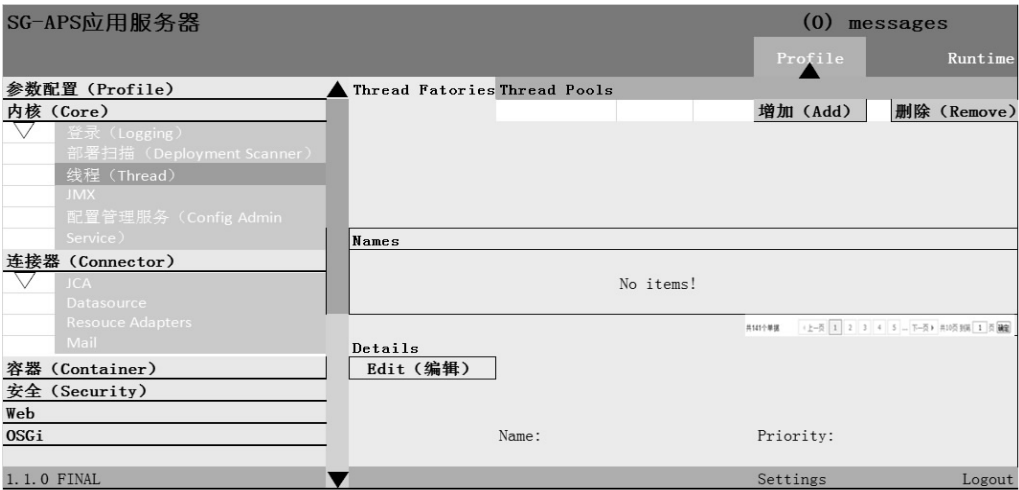


图 2 服务器参数配置界面

(2) 服务器运行监控客户端。

服务器运行监控客户端运行过程中同样通过前述 DomainApiHandler 接口读取 Constants、ConsoleMode 属性获得域管理模式、操作模式和请求参数并构造监控页面,核心方法主要有:

getOperation( String realOperation)//获取操作模式

processRequest( final HttpExchange http)//处理封装自 http 客户端的操作请求

start( HttpServer httpServer, SecurityRealm securityRealm)//启动指定服务器的域服务

stop( HttpServer httpServer)//停止指定服务器

服务器运行监控界面如图 3 所示。



图 3 服务器运行监控界面



(3)集群日志。

集群日志模块是 SG-APS 应用服务器中间件新增的模块,实现了对运行于云平台之上的 SG-APS 应用服务器中间件集群的日志进行统一管理,并由独立运行的日志服务器进行日志收集和分析的功能。集群日志模块由客户端和服务端组成,它们和应用服务器集群之间的关系如图 4 所示。

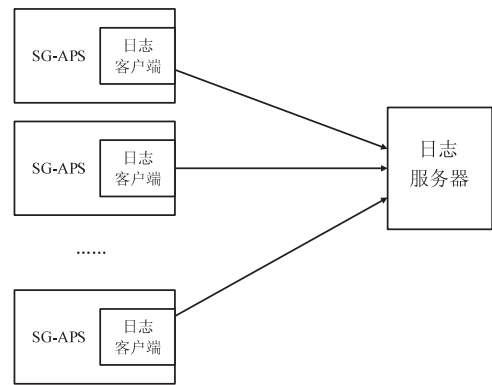


图 4 集群日志

日志服务客户端与 SG-APS 应用服务器主服务运行于同一进程空间,通过预设的日志发送策略向日志服务器发送日志,主要方法包括:

客户端:

getConfig( String tag )//获取给定标识下的配置,包括日志文件路径、读取策略等

readLogFile( File logFile )//读取配置文件

send( Url url, File file )//发送指定日志文件到指定服务器地址

服务端:

saveLog( File logFile )//保存接收到的日志文件

getConfig( String tag )//获取给定标识的配置,如日志文件存放路径、处理策略等

(4)自动运维接口。

以接口的形式提供获取 SG-APS 中 JVM 和内核服务信息的方法,JVM 信息包括:堆内存、非堆内存、垃圾收集、线程、类加载等信息;内核服务信息包括 Web 线程池、EJB 线程、事务处理情况、http 请求应答情况等信息。核心方法包括:

getJVMInfo()//获取 Java 虚拟机信息

getHeapSpace()//获取堆内存空间大小

getJVMGCInfo()//获取垃圾回收信息

getJVMThreadInfo()//获取 Java 虚拟机当前线程信息

getDeployInfo()//获取部署包信息

getEJBThreadPoolInfo()//获取 EJB 线程池信息

getSourceConnectPoolInfo()//获取数据库连接池信息

(5)安装工具。

安装工具用于在 Windows 系统下安装 SG-APS,界面如图 5 所示。



图 5 安装工具

2.3 关键技术

(1)HTTP 请求分发处理。

HTTP 请求分发处理流程如图 6 所示。

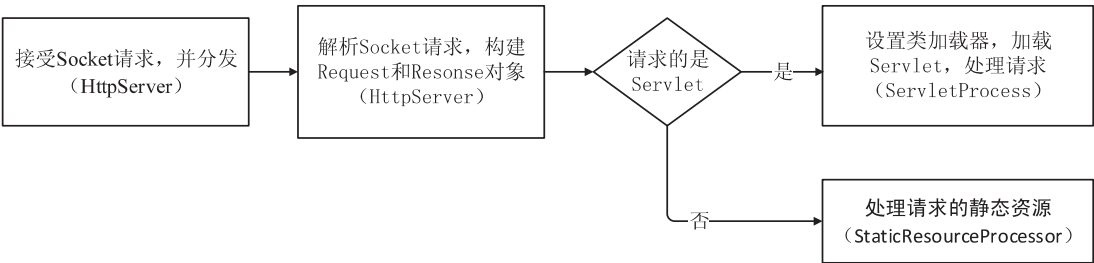


图 6 HTTP 请求处理流程

(2)JSP 编译。

JSP 文件首先被 JSP 引擎(JSP Engine)转换成为一个 Servlet。如果在转换过程中发现 JSP 文件有语法错误则过程中断,并向客户端发出出错信息;如果转换成功,生成的 Servlet 代码被编译为 class 文件并随即被 JSP 引擎加载到内存中。如果 JSP 文件被修改了,应用服务器将自动地对文件重新转换并编译,并用新产生的 class 字节码取代内存中的 Servlet。

(3)线程池数据

线程池是一种多线程处理形式。应用服务器维护一组可重复使用的线程。在任务处理过程中,应用服务器首先将任务添加到队列,然后在线程池选择一个线程启动任务。如果线程池中所有线程都未结束,新进任务将被挂起以等待其他任务结束。

(4)对象序列化。

对象序列化反序列化是将对象实例的状态存储到存储媒体的过程。在随后对对象进行反序列化时,将创建出与原对象完全相同的副本。

### 3 结束语

中国已将信息安全上升为国家战略。为确保作为国民经济要害部门的电力信息系统的“自主可控、安全可靠”,在充分了解和掌握国际标准并参考开源应用服务器软件的基础上,国家电网公司开发了适用于电力信息化的企业级应用服务器 SG-APS。SG-APS 应用服务器中间件共包括一级模块 11 个,二级模块 50 个,三级模块 1 400 余个,涵盖了服务容器、J2EE 核心组件、线程池、数据库连接池、集群管理、安全管理、日志管理、事务服务、监视服务、快照服务和部署服务等核心服务。

SG-APS 应用服务器中间件 1.0 版本的开发工作现已基本完成,正在进行紧密测试,后期将在国家电网信息化系统中逐步替换国外产品。文中介绍了国家电网应用服务器的设计与开发,应用推广效果及后续改进还有待进一步的研究和探讨。

#### 参考文献:

- [1] 刘振亚,张启平. 国家电网发展模式研究[J]. 中国电机工程学报,2013,33(7):1-10.
- [2] 孙丕石,曹占峰,王亚玲,等. 国家电网公司数据交换平台研发与应用[J]. 电网技术,2008,32(22):62-67.
- [3] 韩小涛,尹项根,张哲,等. 嵌入式 Web 服务器技术及其在电力系统中的应用综述[J]. 电网技术,2003,27(5):58-62.
- [4] Cheng A J, Chen Y Y, Huang Y T, et al. Personalized travel recommendation by mining people attributes from community-contributed photos[C]//Proceedings of the 19th international conference on multimedia 2011. Scottsdale, AZ, USA: [s. n.], 2011:291-303.
- [5] Chen Y Y, Cheng A J, Hsu W H. Travel recommendation by mining people attributes and travel group types from community-contributed photos[J]. IEEE Transactions on Multimedia, 2013,15(6):1283-1295.
- [6] 项亮. 推荐系统实践[M]. 北京:人民邮电出版社,2012.
- [7] Ester M, Kriegl H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Proc of 2nd international conference on knowledge

- discovery and data mining. [s. l.]: [s. n.], 1996:226-231.
- [8] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on world wide web. Hong Kong, China: [s. n.], 2001:285-295.
- [9] 邓爱林,朱扬勇,施伯乐. 基于项目评分预测的协同过滤推荐算法[J]. 软件学报,2003,14(9):1621-1628.
- [10] Comaniciu D, Meer P. Mean shift: a robust approach toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002,24(5):603-619.
- [11] Willmott C J, Matsuura K. Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in assessing average model performance[J]. Climate Research, 2005,30(1):79-82.
- [12] 赵子岩,张大伟. 国家电网公司“十二五”电力通信业务需求分析[J]. 电力系统通信,2011,32(5):56-60.
- [13] 2015 年信息化建设实施意见[EB/OL]. 2015. <http://www.sgcc.com.cn/kjcx/xxhgzt/322390.shtml>.
- [14] Kassem N. J2EE 权威指南[M]. 北京:中国电力出版社,2002.
- [15] Implementing enterprise web services[EB/OL]. 2013. <https://jcp.org/en/jsr/detail?id=109>.
- [16] Enterprise JavaBeansTM 3.0[EB/OL]. 2007. <https://jcp.org/en/jsr/detail?id=220>.
- [17] Web services metadata for the JavaTM platform[EB/OL]. 2013. <https://jcp.org/en/jsr/detail?id=181>.
- [18] XML parsing specification[EB/OL]. 2000. <https://jcp.org/en/jsr/detail?id=5>.
- [19] JDBC TM 3.0 specification[EB/OL]. 2002. <https://jcp.org/en/jsr/detail?id=54>.
- [20] JavaTM Message Service (JMS) API[EB/OL]. 2003. <https://jcp.org/en/jsr/detail?id=914>.
- [21] JavaMailTM specification[EB/OL]. 2000. <https://jcp.org/en/jsr/detail?id=904>.
- [22] J2EETM connector architecture[EB/OL]. 2001. <https://jcp.org/en/jsr/detail?id=16>.
- [23] Steel C, Nagappan R. Core security patterns: best practices and strategies for J2EE, web services, and identity management[M]. [s. l.]: Pearson Education, 2006.

(上接第 126 页)

2010:143-152.