

基于云计算 SaaS 应用软件开发模式研究

冀汶莉,李向军,陈夏玉

(西安科技大学 通信与信息工程学院, 陕西 西安 710054)

摘 要: 以 Google App Engine(GAE) 云平台为切入点,对其中的数据存储技术(BigTable)做了深入的分析 and 研究.在此基础上利用 GAE 所提供的开发工具,基于面向对象的思想,采用 Struts+Spring+Jsp 的开发框架,使用了 Google App Engine 开发技术,设计并实现了人事管理的原型系统.通过对原型系统的开发和云端的部署及测试,分析了原型系统的性能.在此基础上,提出了在云平台上开发 SaaS 软件或进行数据迁移需要探索 and 解决的问题,认为在云计算发展的初期,在云端只适合开发部署基于请求-应答的高伸缩性的不涉及多表关联的简单互联网应用系统、简单存储系统及需要高效性计算的智能决策系统.

关键词: 云平台;GAE;BigTable;SaaS

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)07-0137-05

The Research of SaaS Application Development Model Based on Cloud Computing

Ji Wen-li, Li Xiang-jun, Chen Xia-yu

(College of Telecommunication and Information Engineering, Xi'an Science and Technology University, Xi'an 710054, China)

Abstract: This paper starts from the Google App Engine (GAE) and makes deep analysis and study of its data-storage technique (Bigtable). And on this basis, a prototype system is designed and implemented, which Based on the object-oriented thought, using the Struts + Spring+Jsp development framework, using the Google App Engine development technology. By developing the prototype system as well as deploying and testing of the cloud, we analyze the performance of the prototype system. On this basis, put forward the suitable for the characteristics of SaaS software developed by cloud platforms and the problem should pay attention in the development process and present the author's view that, in the early stages of the development of cloud computing, in the cloud is only suitable for development and deployment of high scalability based on request/reply does not involve multiple tables associated with simple Internet application system, storage system and the intelligent decision-making system need high efficiency calculation.

Key words: cloud platform;GAE;BigTable;SaaS

1 引言

云计算是一种新型共享基础构架的计算模式,这种模式简而言之就是按照某种规则将原始数据先进行分割,在分别交由不同的计算机进行处理,最后将处理的结果进行汇总,得到原始数据最终的处理结果.其中共享基础构架是由计算资源(包括服务

器、数据存储)虚拟化来提供的.目前世界上如 Google、IBM、Microsoft、Amazon、Yahoo、Oracle 等很多国际知名 IT 公司,都已涉及并推动云计算技术的发展.

云计算包含 3 个层次的服务,分别为:基础设施即服务(IaaS)、平台即服务(PaaS)和软件即服务(SaaS),不同的云平台提供不同层次的服务^[1],

收稿日期:2013-09-11;修回日期:2013-11-02

基金项目:陕西省科技厅科技攻关计划项目(2012K06-16)

GAE(Goole App Engine)云平台是平台即服务(PaaS)的典型代表。

SaaS(软件即服务)模式是一种新的软件开发模式,从技术角度上看它是一种在互联网中使用浏览器来实现应用的应用模型,软件是以托管的方式部署在云平台上.但它并不是开发技术上的革新,而是业务模型的革新,它的开发目前仍然可以遵循继续发展的基本 Web 标准,可以使用任何目前基于 Web 的开发模式开发具备 SaaS 特点的应用软件。

但是否所有的应用系统都可以迁移或重新开发部署到云平台,这与云平台提供什么模式的数据存储技术有密切的关系.本文在此基础上,以 GAE 云平台为例,深入抛析了 BigTable 的数据存储技术,并基于 GAE 设计开发了一个原型系统。

2 GAE 体系架构以及核心技术

云计算重要的表现形式是计算资源服务化,按需为用户提供弹性资源是它的重要特点.一般来说云计算的体系架构可分为服务管理、安全管理、用户访问接口 3 层^[2],如图 1 所示。

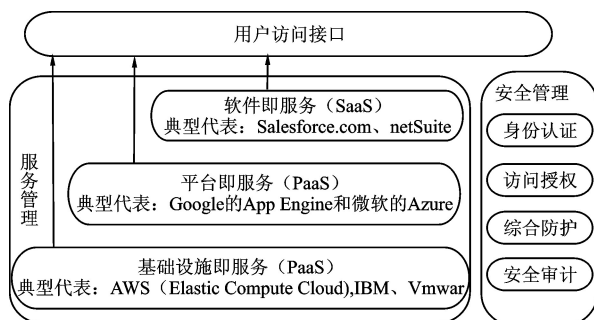


图 1 云体系架构

Google 是全球最早提出“云计算”构想的公司之一,下文着重点分析了 Google 云平台与应用程序开发密切相关的文件系统、数据存储等核心技术。

2.1 Google 文件系统 GFS

Google 文件系统 GFS(Google File System)为 Google 搜索引擎提供了海量数据的分布式存储,它有 4 个方面的特点:(1)当大量节点失效时的软件容错性;(2)文件系统支持 G 级别的海量存储;(3)支持文件以追加方式批量写入;(4)支持多个客户端对一个文件的高并发性的写入操作. GFS 采用大量廉价的服务器来构建分布式文件系统的物理层,这是 GFS 相比于其他分布式文件系统的创新点(如:IBM 的 GPFS, Sun 的 Lustre). 同时 GFS 在容错处理上,将容错任务交给文件系统来处理,利用软件的方法来解决系统的性能问题,极大降低了数据存储

成本^[3]. GFS 的整体架构如图 2 所示。

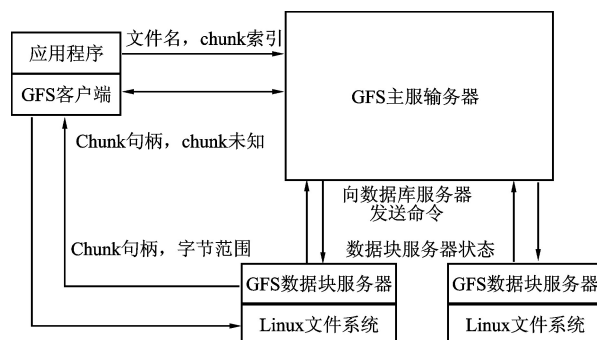


图 2 GFS 整体架构

从图 2 可以看出, GFS 是以 GFS 集群的形式存在,一个 GFS 集群是由一个主服务器(Master)和多个数据块服务器(Chunk Server)组成的,若干个客户端可以连续频繁访问这些具有数以千计的节点和超大存储空间的 GFS 集群. 主服务器是 GFS 系统的管理节点,在逻辑上只有一个,主要用来存放元数据,包括文件命名空间、文件到块的映射、文件访问控制、块的物理位置等信息. 数据块存服务器是用来存储具体的数据,数据以文件的形式按照固定的大小进行分割成块(默认大小是 64 MB),每一个块称为一个 Chunk 以 Linux 文件的形式存储在本地硬盘上. 每一个 Chunk 都有相应的一个索引号,Client 通过 Master 可以获取这个索引号^[4-5]。

客户端访问 GFS 时,首先通过文件名和 Chunk 索引在主服务器获取相应的数据块存服务器的句柄和位置信息. 根据数据块存服务器的位置信息来完成数据的存取. 在整个交互的过程中,客户端和主服务器之间并没有产生大量的数据流,而真正的数据流发生在客户端和数据块存服务器之间,这使得主服务器的负载极大的降低. 同时,由于数据块存服务器上的文件以块的形式存放,使得客户端同时可以访问多个数据块服务器,提升了系统高度并发性。

2.2 Bigtable

作为云计算的支撑技术之一,云需要某种存储模式来保存虚拟机映像、用户的应用和应用所需的持久化数据,因此在云中也具有数据库的概念. Google 采用 BigTable 作为它的云数据库, BigTable 是“键-值”的 NoSQL 结构的数据模型的典型代表,它的首要任务是保证 Google 快速搜索所涉及的海量数据. BigTable 是构建在 GFS 之上的一个分布式存储系统,本质上它是稀疏的、分布式的、持久化存储的多维度排序 Map. 它的物理存储模式采用了类似数据结构中 B+ 树索引组织表的模式,称为多维

度的分类映射,其中的分支和叶节点可以分布在多台机器上,通过在成千台计算机上分布数据,提供了对 PB 级数据的快速访问.由于节点是分布式的,随着树的生长节点会拆分,根据应用需求量获得高伸缩性成为可能.

BigTable 将数据统一看作无意义的字符串,客户端需要将结构化和非结构化的数据串行化再存入 BigTable. BigTable 中的数据是通过一个行关键字(Row Key)、一个列关键字(Column Key)以及一个时间戳(Time Stamp)进行索引的. BigTable 对存储在其中的数据以字符串进行存储,不做任何解析. BigTable 的存储逻辑可以表示为

```
( row: string, column: string, time:
int64)string[6]
```

其中行是第一级索引,行键是大小在 10~100 字节之间的任意字符串.虽然 Bigtable 不支持一般意义上的事务,但行中数据的读写是原子性的. BigTable 是按照行键的字典顺序进行数据维护. BigTable 中的数据会随着行键的增大动态划分为片(tablet),tablet 是负载均衡和数据分发的基本单元,大小在 100 MB 到 200 MB 之间,存储在不同的片服务器上.

列是 BigTable 的第二级索引,列会被划分为多个列族(column family),列族是基本的访问控制单元,列族中的数据类型是相同的.列族中的列是可以任意的变化的,列键是以 family:qualifier 的格式命名的.family 是列族的名称,qualifier 是列族中的列的名称^[7].时间戳是 BigTable 中的第三级索引. Bigtable 中的数据可以按照时间戳来保存成多个版本.时间戳的具体值可以由 BigTable 自身进行赋值,也可以由客户端指定特定的值,BigTable 的时间戳是 64 位整数.数据的不同版本是按照时间戳的降序顺序来存储的,所以系统先会读到最新的数据.

下面是结合本文中的开发实例给出的一个完成的 BigTable 数据模型,描述了人事管理系统中用户实体的数据,如图 3 所示.

其中 Table 表示一张表,row 表示 1 行的数据.userName,sex 为该行的列,address 为列族包括两列:personAdress,officeAddress,其中每一列保存不同的版本,以时间戳来区分.图 3 中给出的例子共有两行数据,伴随着行的增多,Table 会根据行键自动划分为 tablet,这些 tablet 会分布都不同的计算机上,tablet 中的数据最终会存储到 GFS 文件系统中.

```
Table{
  "row":{"userName","sex","address">{
    "address:personAdress":{
      "personAdress1":"20130906123623",
      "personAdress2":"20130905123623"|
    }
    "address:officeAddress":{
      "officeAddress1":"20130906123623",
      "officeAddress2":"20130905123623"
    }
  }},...
}
```

图 3 Bigtable 数据模型

3 基于 GAE 构建部署原型系统

GAE 允许用户在 Google 的平台上开发自己的 web 应用,并可以将其应用部署到 Google 云平台之上.对用户来说,不需要购买和搭建昂贵的硬件和软件系统,只需要关注系统本身的业务处理.当前 GAE 支持 Python 语言、JAVA 语言和 Go 语言.作者使用了目前流行的 J2EE 的基本开发模式设计开发了人事管理的原型系统.原型系统是目前正在开发的基于公共云平台的 OA 项目的一个较为独立的子模块,研究如何构建部署在 GAE 上的原型系统可以作为在公共云平台实现 OA 系统的前期预研.

3.1 原型系统应用设计

Google 提供的云平台对于开发者仍然支持面向对象的设计思想,原型系统在设计阶段包含两个实体:员工和部门实体,如图 4 所示.员工跟部门之间并无强制性的对应关系,在员工实体中存放部门 Id 属性,这样通过部门 Id 可以查询到属于该部门的员工,通过员工 Id 可以查到该员工所属相关部门.同时,员工可以不属于任何部门,脱离部门而存在.上述对象的建立实际是由 Google 的 BigTable 的数据存储特征决定的.关于 BigTable 的设计及存储原理在前文以作详细论述,这里就不在赘述.

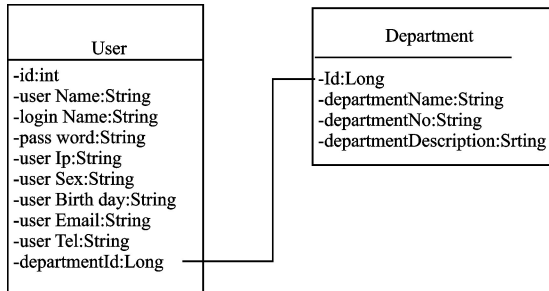


图 4 部门员工实体图

GAE 是由数据存储区(DataStore)对数据进行持久化,DataStore 是基于 BigTable 进行设计的。DataStore 中存储的数据对象称为 entity,一个 entity 有多个属性,每个 entity 都有一个 key 唯一标识这个 entity(类似关系数据库的 id)^[8]。上述的 User 和 Department 对象就是 DataStore 中的两个 entity,可以和关系数据库模型的表相对应。

GAE 支持两种标准接口分别是 Java Data Object(JDO)以及 Java Persistence API(JPA),用于将对象持久化到数据存储区(DataStore)。这两种接口都允许开发者用类型安全的 Java 类来描述数据结构^[9]。在原型系统使用 JDO 来实现数据实例化的功能。在 JDO 中,通过 PersistenceManager 类实例化对象、更新对象、取出以及查询对象^[10]。PersistenceManager 实例是由 PersistenceManagerFactory 对象提供的。如下示例程序所示。

```
package com.jlsoft.oa.service.user;

import javax.jdo.JDOHelper;
import javax.jdo.PersistenceManagerFactory;

public class PMF {
    private static final PersistenceManagerFactory pmfInstance = JDOHelper
        .getPersistenceManagerFactory("transactions-optional");

    private PMF() {
    }

    public static PersistenceManagerFactory get() {
        return pmfInstance;
    }
}
```

在将具体对象持久化的时候,先通过 PMF 获取 PersistentManager 的对象,然后调用相应的方法对数据做保存,删除,修改,查询等处理。如下示例程序所示,完成了对 User 对象的持久化。

```
public class UserDao {
    public void save(Long departmentId){
        PersistenceManager pm = PMF.get().getPersistenceManager();
        try {
            pm.currentTransaction().begin(); //事务开始
            User user = new User();
            user.setLoginName("test");
            user.setPassword("test");
            user.setUserBirthday("1988-02-14");
            user.setUserEmail("test@qq.com");
            user.setUserName("test");
            user.setDepartmentId(departmentId);
            pm.makePersistent(user); //持久化
            pm.currentTransaction().commit(); //调用commit, 事务结束
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } finally{
            pm.close();
        }
    }
}
```

3.2 部署与实现

Google 在 2009 年发布的 Google App Engine For Java 支持大多数的 J2EE 框架在 GAE 环境中搭建 Web 应用。作者采用当前流行的 Struts2.0+Spring+Jsp 开发模式,在 GAE 环境中实现了一个人事管理的原型系统,系统的功能主要包括员工管理以及部门管理。

在 GAE 环境中开发应用程序时,首先需要创

建 Google 账户,并下载 App Engine SDK(Google 提供的 Eclipse 插件)。插件正常安装并简单配置之后就可以进行应用程序开发。在 GAE 中部署应用程序主要包括以下几个步骤:创建域名、开发应用程序、上传应用程序、检测统计等。在 GAE 下系统最终的运行结果如图 5 所示。



图 5 系统实现截图

4 系统性能测试

本次测试是将系统放在企业内部网的普通环境下和 GAE 环境下分别测试,其中普通环境下的数据库采用的是 MySQL。主要包括数据存储,读取速率的测试以及系统性能的测试。在进行数据存储、读取速率测试时,将测试同时插入以及读取 100、1000、10000 条数据的所需时间。性能测试主要测试 ART (AverageResponseTime, 平均响应时间)和 TPS (RequestsPerSecond, 每秒可相应的请求数)。

测试结果如表 1~表 2 所示。

表 1 普通环境数据读写速度

数据操作	100	1 000	10 000
读速度/ms	56	421	2 560
写速度/ms	1 294	18 009	>203 484

表 2 GAE 环境数据读写速度

数据操作	100	1 000	10 000
读速度/ms	89	650	4 569
写速度/ms	31	54	383

从表 1 和表 2 可以看出,GAE 环境中 BigTable 的存储速率远远大于普通环境下数据的存储速率,并且随着数据量的增大,这种优势会越来越大。但是 GAE 环境中数据的读取速率小于普通环境下数据读取的速率,这与 BigTable 的设计有关,BigTable 并不完全支持关系数据库,在数据的查询中没有像关系数据库那样对查询做了优化,如果查询涉及多表之间的关联,效率会更低。如果是简单的 web 应用或存储系统,由于 Web 系统中大数据量的获取可以通过分页技术实现数据的分批加载,因此可以忽

略 BigTable 读取数据速率方面的不足。

表 3 中描述了在 GAE 环境下系统的性能测试结果。两组测试数据的 ART 结果,符合 Web 压力测试标准,可见系统的性能良好。

表 3 GAE 环境下系统压力测试

次数 \ 参数	第一次	第二次
点击数	20 832	50 000
ART/ms	5.75	1.18
TPS	174	560
成功连接数	20 832	50 000

从上面的分析可以看出,GAE 环境下的 Web 应用在数据的存储以及性能方面均高于普通环境中的 web 应用。但通过测试结果分析可知,GAE 环境下的 Web 应用在数据读取方面相对于传统的基于关系数据库的应用系统并无明显的优势。

5 结束语

本文以 GAE 云平台为例具体分析了 Google 的 GFS,BigTable 的核心技术,并基于 Google App engine 开发了原型系统,对原型系统的数据存储速率,系统性能进行了分析。

通过对原型系统的开发和系统测试分析可知,开发 SaaS 模式的应用系统或进行企业基于事务处理系统的数据迁移,需要解决以下问题:

(1)诸如 GAE 这类公有云平台并不支持关系数据库。目前云平台的数据存储多以 NoSQL 模式为主,不能很好的支持多表之间的关联查询。如果应用系统是基于联机事务处理,在设计阶段涉及多表,并且多表之间有较复杂的关联关系。如果不深入研究并实现事务处理和云存储之间的转换机制,直接利用传统的 Web 模式进行 SaaS 服务模式的软件开发或向云端迁移,是无法体现云计算的系统性能优势,甚至会导致比传统的基于关系数据库的联机事务处理系统的性能要差很多。

(2)数据迁移方法问题。企业经过长期的信息化建设,在本地数据库存储了大量数据,很长一段时间应该是本地数据库和云端数据共存的局面。如何实现本地数据库和云端数据有效融合、统一管理;在进行数据迁移时的有效方法也是企业在进行云计算应用前应该深入研究的问题。

(3)数据安全的问题。选择了云平台就意味着某个应用可以使用云端的企业数据,数据的安全性问题将会是一个让企业担忧的问题。因此涉及企业敏

感数据的应用是否应该部署或迁移到云端以及迁移方法都是一个值得研究和探索的问题。

(4)目前各大云平台并无统一的标准,可能存在应用系统被某个云平台套牢的风险。如果建立自己的私有云,则面临私有云的规模小于公有云;组织内部的遗留系统无法轻易的迁移到云中;私有云也无法给用户提供更比现有公有云更好的服务和安全保障等问题。

通过对原型系统的设计开发和测试分析,作者认为在目前云计算技术发展的初期不适合在云端开发或迁移的应用系统是原有的组织内部构建在专有软硬件平台上的遗留系统;涉及实时数据的系统及关键任务的应用;涉及处理机密数据的应用。目前在云端适合开发部署一些基于请求一应答的高伸缩性的简单互联网应用系统;一些非战略的不涉及多表关联查询的应用系统。对于企业大型系统也可以分离一些相对独立的不涉及组织机密数据的子系统部署于云端,目的是高效使用云端强大的数据存储及备份功能。

参考文献:

- [1] 解绍词,葛君伟. 基于 MDA 的云计算软件开发平台模型的研究[J]. 微电子学与计算机,2011,28(9):96-97.
- [2] 刘鹏. 云计算[M]. 北京:电子工业出版社,2011:15-17.
- [3] 王鹏. 云计算的关键技术与应用实例[M]. 北京:人民邮电出版社,2010:37-50.
- [4] Gonzalez H, Halevy AY, Jensen CS, et al. Google fusion tables: Web-centered data management and collaboration[C]//Proc of the ACM SIGMOD Int'l Conf on Management of Data. New York: ACM Press,2010.
- [5] 陈康,郑纬民. 云计算:系统实例与研究现状[J]. 软件学报,2009,20(5):1337-1348.
- [6] Yoon JP. Access control and trustiness for resource management in cloud databases[C]//Proc of the Int'l Conf on Grid and Cloud Database Management. Berlin: Springer-Verlag, 2011:109-131.
- [7] 林子雨,赖永炫,林琛,等. 云数据库研究[J]. 软件学报,2012,23(5):1148-1166.
- [8] Scott Granneman. Google Apps 应用完全指南[M]. 李强,等译. 北京:机械工业出版社,2009:35-48.
- [9] Google App Engine for Java,第3部分:持久性和关系[EB/OL]. [2009-09-14]. http://www.ibm.com/developerworks/cn/java/j-gaej3.html?S_TACT=105AGX52&S_CMP=tec-ccid

(下转第 146 页)

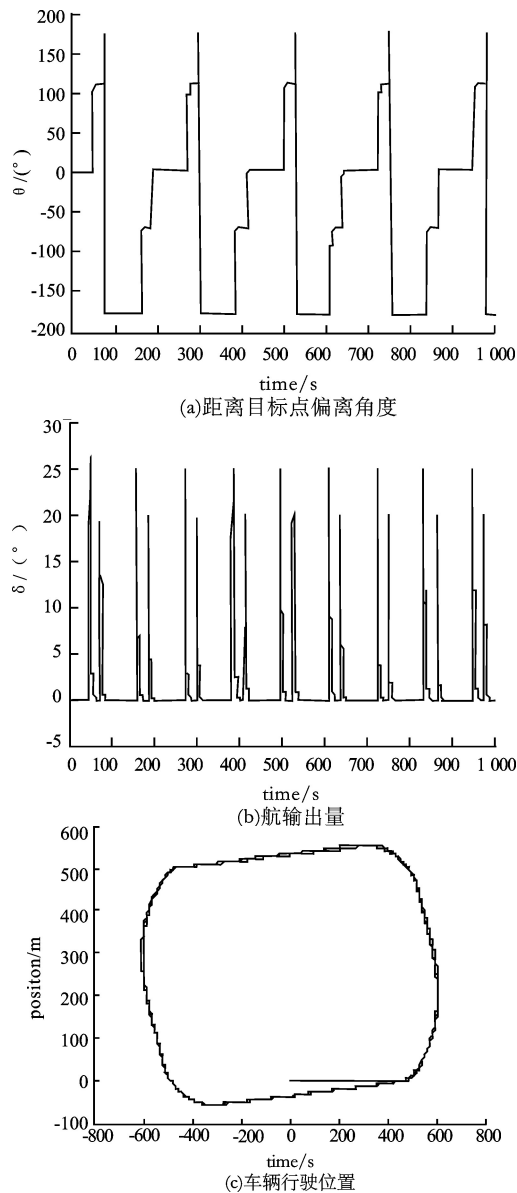


图8 控制仿真结果

tional control with a fractional order $pd\mu$ controller [J]. Journal of Shanghai Jiaotong University, 2008, 9 (6): 679-685.

- [3] Choomuang R, Afzulpurkar N. Hybrid kalman filter/fuzzy logic based position control of autonomous mobile robot[J]. International Journal of Advanced Robotic Systems, 2005, 2 (3): 197-208.
- [4] 宗长富,潘钊,胡丹,等. 基于扩展卡尔曼滤波的信息融合技术在车辆状态估计中的应用[J]. 机械工程学报, 2009, 45(10): 272-277.
- [5] Guivant J, Nebot E, Baiker S. High accuracy navigation using laser range sensors in outdoor applications [C]//Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco:IEEE,2000.
- [6] Bevly D M, Ryu J, Gerdes J C. Integrating INS sensors with GPS measurements for continuous estimation of vehicle sideslip, Roll, and Tire Cornering Stiffness[J]. IEEE Transactions On Intelligent Transportation Systems, 2006, 7(4): 483-493.
- [7] Sotelo M A. Lateral control strategy for autonomous steering of Ackerman-like vehicles[J]. Robotics and Autonomous Systems, 2003, 1016(10):223-233.
- [8] Scheduling S, Dissanayake G. An experiment in autonomous navigation of an underground mining vehicle[J]. IEEE Transactions on Robotics and Automation, 1999, 15(1): 85-95.

作者简介:

王 萌 男,(1988-),硕士研究生. 研究方向为嵌入式系统设计、飞行控制.

史忠科 男,(1956-),博士,教授,博士生导师. 研究方向为飞行试验、飞行控制.

- [2] Zhuang D J, Yu F, Lin Y. Evaluation of a vehicle direc-

(上接第 141 页)

- [10] 胡洁萍,杨树林. JPA 与 SSH 架构整合方案及应用研究[J]. 北京印刷学院学报,2012(6):51-52.

作者简介:

冀汶莉 女,(1973-),硕士,副教授. 研究方向为云计算、物联网技术.

李向军 男,(1988-),硕士. 研究方向为面向服务的软件开发模式、SaaS 软件开发的模式.

陈夏玉 女,(1988-),硕士. 研究方向为数字通信与个人通信.