

Design and Implementation of Forage Production Machinery and Equipment Data Sharing Platform Based on NoSQL Database

Po Su¹, Hui Li¹, Hongqian Chen², Chao Li¹, Ze Li¹, Chaoying Meng^{1*}

¹College of Information and Electrical Engineering, China Agricultural University, Beijing

²Network Center, China Agricultural University, Beijing

Email: *mcy@cau.edu.cn

Received: Apr. 19th, 2013 revised: Apr. 30th, 2013; accepted: May 10th, 2013

Copyright © 2013 Po Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: The forage production machineries and equipments have many different types and the different type owns its special parameters. It is lack of scalability and versatility when building data share platform by relational database, which is not conducive to the information needs of the development of grass industry. Based on non-relational database MongoDB and the characteristics of the forage machinery complex data format, it discussed how to use MongoDB database and data organization form. By using the web.py lightweight framework to develop the web application solve the forage mechanical data interactive access and other practical problems, and realize the sharing of forage production machinery and equipment data.

Keywords: Data Share Platform; Python; MongoDB; Web Application

基于 NoSQL 数据库的饲草料生产机械设备数据共享平台的设计与实现

苏 坡¹, 李 辉¹, 陈红茜², 李 超¹, 李 泽¹, 孟超英^{1*}

¹ 中国农业大学信息与电气工程学院, 北京

² 中国农业大学网络中心, 北京

Email: *mcy@cau.edu.cn

收稿日期: 2013 年 4 月 19 日; 修回日期: 2013 年 4 月 30 日; 录用日期: 2013 年 5 月 10 日

摘 要: 饲草料机械设备数据类型较多, 设备参数复杂, 应用传统关系型数据库构建的数据共享平台缺乏扩展性和通用性, 不利于牧草行业信息化发展需求。因此在研究非关系型数据库 MongoDB 的基础上, 针对饲草料机械数据格式复杂的特点, 对如何应用 MongoDB 数据库及数据组织形式进行讨论, 并利用 Python 语言的开发 web 应用的轻量框架 web.py 进行饲草料机械设备数据共享平台的开发, 解决了饲草料机械数据交互式访问等实际问题, 实现了饲草料生产机械设备数据的共享。

关键词: 数据共享平台; Python; MongoDB; Web 应用

1. 引言

饲草料生产机械设备数据是影响牧草产业信息化发展的主要因素, 该数据格式复杂, 应用关系型数据库会产生存储效率和查询效率低的问题, 同时基于关系型数据库构架的数据共享平台可扩展性差。通过非关系型数据库 MongoDB 实现饲草料生产机械设备数据共享, 解决了现阶段饲草料生产机械设备数据获

数据库会产生存储效率和查询效率低的问题, 同时基于关系型数据库构架的数据共享平台可扩展性差。通过非关系型数据库 MongoDB 实现饲草料生产机械设备数据共享, 解决了现阶段饲草料生产机械设备数据获

取难的问题,满足饲草产业规模的不断扩大以及牧草种植者对于先进机械的迫切需求,使得牧草机械研发机构和牧草生产者之间交流变得越来越频繁;可以提高科研单位对市场需求的敏感度,从而对急需的机型进行开发;可以提高机械用户对市场的了解程度;多方面促进牧草产业发展从而带动社会发展和经济建设^[1]。

本文以公益性行业项目为依托,着重研究饲草料机械设备数据特定和应用需求。本文首先分析非关系型数据库等计算机技术在网络应用、大规模数据处理的理论意义和应用方法;然后,以此为理论基础,结合实际的项目需求和开发过程,介绍饲草料机械设备数据平台的开发、应用等实际问题的解决方案,为牧草行业的机械设备数据的共享提供一个成熟可行的应用实例。

2. 技术支持

2.1. NoSQL

NoSQL 是区别于传统关系型数据库的数据库管理系统的统称。与统治数据库世界三十年余年的关系型数据库相比, NoSQL 最大的不同是不使用 SQL(结构化查询语言)作为查询语言,其数据存储可以不依照固定的表格模式,通常具备水平可扩展的特征^[2]。

“管理基于互联网的大数据”,是 NoSQL 数据库一个重要的应用场景。由于大数据具备数据量大、多结构化、增长速度快和价值密度低的特点,采用传统关系型数据库管理技术往往面临成本支出过多、扩展性差、数据快速查询困难等问题。NoSQL 数据库可以在很大程度上满足企业对大数据的抓取、管理和处理需求。也正是基于这样的数据管理需求, NoSQL 在互联网企业中得到了非常广泛的应用。伴随 Web 2.0 时代汹涌而来的大数据(也称为“海量数据”)让 NoSQL 更进一步走到台前^[3]。

饲草料生产机械设备数据包括机械本身数据信息、生产厂家信息和适用情况信息等,数据信息量大;应用 NoSQL 可以很好的适应饲草料生产机械设备数据量大的特点,同时还可以方便地进行扩展。

2.2. MongoDB 数据库

MongoDB 是一个高性能、开源、无模式的文档

型数据库。它是一个介于关系数据库和非关系数据库之间的产品,是非关系数据库当中功能最丰富,最像关系数据库的。它支持的数据结构非常简单,是类似 json 的 bson 格式,因此可以存储比较复杂的数据类型。MongoDB 最大的特点是它支持的查询语言非常强大,其语法有点类似面向对象的查询语言,几乎可以实现类似关系数据库表单查询的绝大部分功能,而且还支持对数据建立索引^[4]。

MongoDB 功能非常强大:适合存储对象及 json 形式的数据库;支持丰富的查询表达式,查询指令使用 json 形式的标记,可以轻易查询文档中内嵌的文档或数组;高效的传统存储方式,支持二进制数据对象,例如图片或视频文件。

MongoDB 特别适用于实时的插入,更新与查询,并具备网站实时数据存储所需的复制及高伸缩性;由于 MongoDB 的性能很好,亦可以作为信息基础设施的缓存层,如在系统重启后,通过 MongoDB 搭建的持久化缓存层可以避免下层的数据源过载^[5]。

饲草料生产机械设备数据的格式复杂,采用 MongoDB 的基于文档的存储结构可以将关系型数据库中的多张表结构简化成一种文档模式。这种文档模式不是固定的格式,因此可以进行扩展;同时 MongoDB 提供了多种类似关系型数据库的查询语言,可以通过一条查询语句解决在关系型数据库中需要连接操作的查询语句。

3. 共享平台功能与设计

饲草料机械设备数据共享平台是一个面向互联网发布牧草设备数据信息的公共服务网站,通过网站为科研人员、企业用户以及普通公众发布机械信息。

平台的数据来源有饲草料机械设备数据,也有生产厂家数据等数据。将采集到的饲草料机械数据进行加工、处理、存储,然后借助 JavaScript 和 JQuery 库可以对数据进行丰富的信息展示和实现多样的功能,从而形象而直观地为广大用户提供信息服务。运用多种方式实现导入数据,通过高效的数据组织形式进行数据库设计,从而优化数据库的管理、更新、维护和更加快速的查询功能。采用多种形式展现所查询出的饲草料机械数据信息,从而能够更加有效地促进饲草料行业的快速稳定发展。

该共享平台主要包括四大功能模块，饲草料机械数据导入模块、饲草料机械数据共享模块、公告政策管理模块、系统管理模块，图 1 为详细的功能模块划分。

饲草料机械数据导入模块为系统提供数据来源。普通管理员主要完成饲草料机械数据的添加、修改和删除功能。普通管理员首先需要将相关机械数据录入到数据库中，可以采用手动录入和文件导入的方式；手动录入就是普通管理员点击添加数据按钮之后，逐项添加需要录入的信息，适合数据量较少的情况；文件导入方式是普通管理员点击导入 excel 文件按钮之后，通过上传 excel 文件，一次添加大量数据，该方法适合数据量比较大，手动录入方式已经不再适用的情况。超级管理员主要任务是对添加的数据进行审核，审核未通过时，需要普通管理员重新录入新的数据，审核通过之后，该数据就可以被检索到。同时超级管理员还有创建用户和修改权限的权利。

饲草料机械数据共享模块主要针对普通用户完成整个数据共享的流程和各项信息的查看。普通用户第一步是确定数据检索的策略，一共有四种检索策略，根据机械种类检索机械设备、根据功能检索机械设备、根据生产厂家检索机械设备和高级检索，在选择完检索策略之后，输入相关检索条件之后，提交。普通用户会得到该检索条件的结果列表，列表展示该类机械的一般信息，点击一条机械数据的查看详细之后，会得到该条机械数据的详细信息，包括机械特性数据、图片数据、视频数据、生产厂家数据等。

公告政策管理模块主要针对超级管理员和普通管理员，主要完成饲草料机械共享数据平台门户新闻等内容管理功能。超级管理员或普通管理员登录管理平台，在文章管理栏目中选择需要更新的文章类型，如行业资讯、政策法规等，点击添加按钮之后，出现添加文章信息的页面，在该页面上填写文章标题、添加日期以及文章的具体内容和图片信息，最后点击提交，添加到数据库之中。同时超级管理员或普通管理员还可以对文章进行删除和修改操作，具体步骤类似于添加操作。

系统管理模块主要针对超级管理员，主要完成对系统用户添加或权限修改的管理功能。超级管理员可以利用该模块，通过登录管理平台，选择系统管理栏

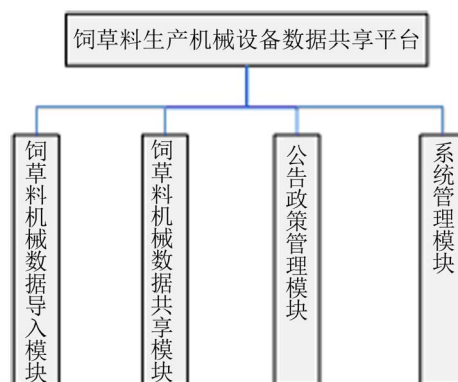


Figure 1. The function module structure's diagram of the forage machinery data sharing platform
图 1. 饲草料生产机械设备数据共享平台功能模块结构图

目，在相应的页面中选择创建用户或者更改权限。

4. 共享平台开发与实现

4.1. 共享平台体系架构

该项目平台采用的是 python 语言的 web.py 框架进行开发，采用与现在流行的 MVC 架构相对的 MTC 架构，MTC 即是 Model 层、Template 层以及 Controller 层；Model 层主要是有关数据 CRUD 操作；Controller 层主要是关于系统的业务流程；Template 层主要是用来进行展示用，web.py 框架通过模板技术，简化表现层的开发，同时还可以支持多种模板替换引擎^[5]。

系统结构的设计结合了 B/S 模式、Ajax、python、MongoDB 以及各种开源框架，层次结构如图 2，分别为：表示层、业务逻辑层、数据层和数据资源层。

1) 表示层是用户界面层，WebUI 采用 ExtJS 和 JQuery 类库进行开发，通过 Ajax 与服务器端交互，具有很强的交互性和用户友好性。用户通过 Web 浏览器来与服务器进行交互，获得自己所需的信息及实现饲草料机械数据管理、公告政策管理等功能。

2) 业务逻辑层是系统的核心层次，完成系统的所有业务功能，主要包括 Web 服务器，同时还包括数据共享管理所需要的专业化的服务和构件，包括：机械数据查询共享构件、用户权限构件以及其他管理构件。

3) 数据层是对底层数据库的抽象，是一个专注于数据操作的逻辑层面，向业务逻辑层提供对底层数据库访问的统一的接口。

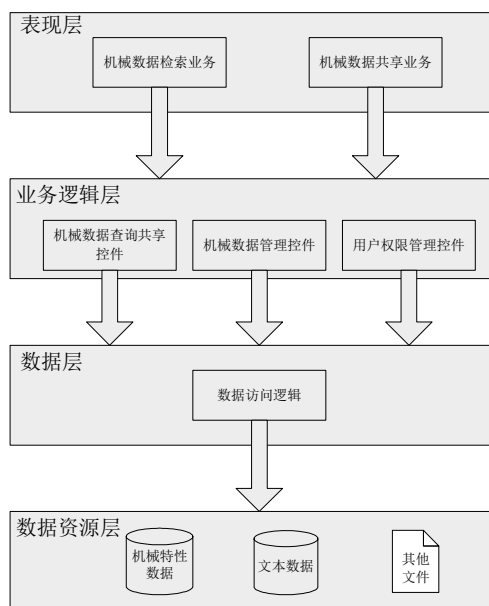


Figure 2. The structure's diagram of sharing platform
图 2. 平台体系结构图

4) 数据资源层集成各种异构数据源向上层提供数据，是数据承载的实体。包括机械信息数据库、其他信息数据库以及各类文件。

4.2. 共享平台数据存储机制

饲草料生产机械设备数据共享平台是一个针对科研人员、社会用户和企业的共享平台，因此确保饲草料生产机械数据正确、稳定和快速地存储是重中之重。传统关系型数据库在存储饲草料机械数据，由于不同机械类型数据的字段不全相同，所以需要针对每个机械类型建表，如此一来，使得访问数据库的接口操作不同，需要操作不同的表；但若要是将数据库访问接口统一，则需要对上述不同种类的机械数据字段进行统一，统一建立一张机械数据的大表，该表需要包含不同机械数据的所有字段；简化数据接口的后果是在存储不同类型的机械数据时，会产生大量的空字段，导致存储的效率差^[6]。

MongoDB 与关系型数据库的不同之处体现在：不需要预定义表结构，所以的实体都是以文档的形式来描述，文档没有模式之分。在饲草料生产机械设备数据共享平台中，每类饲草料机械数据所拥有的字段结构不同，在使用 MongoDB 进行数据库设计时，不用受到模式的限制，可以将通用的字段设计形成一个文档，将区别于别的饲草料机械数据的字段设计成为

上述文档的一个属性，这样就不会出现关系型数据库存储效率低的问题，同时这数据可以组织得更自然。

本文所涉及的饲草料生产机械设备数据共享平台的数据主要通过上传 Excel 文件来维护机械数据，后台语言采用 python，数据库采用 MongoDB。Excel 上传维护机械数据包括读取控制、校验控制和错误提示三方面内容；读取控制是指在读取 Excel 为了防止重复添加机械数据而进行的控制；在读取 Excel 时，首先会维护一个列表，该列表包含现在数据库中所包含的机械数据的一级、二级类型名称和机械型号组成的字符串；Excel 每行数据代表一个特定类型的机械数据，在读取每一行数据时，首先会将该行机械数据的一级、二级类型名称和对应型号组成字符串，然后判断该字符串是否在初始化时维护的列表中，若存在于列表中，则说明当前数据库中已经包含该条机械数据，不需要进行多余的添加操作；若不存在于列表中，则说明当前数据库中没有该条机械数据，则需要进一步地添加操作。校验控制是为了防止读取 Excel 的数据出现类型错误，如需要数字表示的字段错误地填写成字符串。校验控制和错误提示同时进行，都是在读取控制之后，采用 python 中的 map 类型保存着 Excel 中不同位序数据的对应的数据类型；在循环读取 Excel 一行时，会根据 map 中保存数据，对读取到的数据进行转化，若是转化失败，将出错的行和列填写到错误列表中；若转换成功，则用 Excel 第一行数据作为 key 值，每行数据的值作为 value 值，从而形成一行机械数据对应的一个 map，这个 map 和 MongoDB 中的 BSON 数据格式类似，直接调用数据库访问接口，将这个 map 插入到数据库中^[7]。通过 MongoDB 数据库本身数据插入速度快的优势和三种为了确保数据正确性的控制可以确保饲草料生产机械数据正确、稳定和快速地存储^[7]。

为了验证上述论断，设计了下面三种常用的验证方法。具体环境为同一电脑主机，MongoDB 1.4 和 MySQL 进行效率比较。三种验证方法分别为：

- 1) 插入 1000 条记录数据。
- 2) 以非主键字段进行 1000 条记录数据的更新。
- 3) 以主键字段进行 1000 条记录数据的更新。

在表 1 中，可以看出，MongoDB 在插入数据和以非主键更新数据的表现较 MySQL 有很大的优势，

Tables 1. The efficiency of MySQL and MongoDB
表 1. MySQL 和 MongoDB 的效率

	MySQL	MongoDB
插入 10,000 条数据	16,327 ms	1912 ms
以非主键更新 1000 条数据	31,151 ms	6484 ms
以主键更新 1000 条数据	1592 ms	1097 ms

在以主键更新数据的表现上略胜 MySQL 一筹。上述结果可以验证 MongoDB 可以提高数据录入的效率这一结论^[7]。

4.3. 系统实现

系统完全基于 MTC 模式，将业务逻辑、显示逻辑和数据逻辑完全解耦，有利于系统的维护和扩展。在 WebUI 方面，采用 ExtJS 和 JQuery 框架，增强了系统的交互性和展现能力。在 Web 应用开发框架，选用 web.py 进行开发，可以方便的与其他框架进行集成。由于该系统涉及大量的机械特性数据，因此在数据库管理系统方面选用 MongoDB，利用其强大的数据管理功能管理系统的大量数据。开发环境选择了 Aptana Studio 3, 主要使用 python 和 JavaScript 完成开发工作。Web.py 框架使得控制逻辑可以很清晰地分离出来^[8]。

```
import web
urls = (
    '/manager/updateList', 'updateList'
)
class updateList:
    def POST(self,_id):
        data = web.input()
        name = data['name']
        count = getCount(_id)
        if count! = 0:
            if modifyName(_id,name)and modify-Text
(_id,name):
                message = { "msg": "操作成功", "su
ccceed": True ,"success": True}
                return json.dumps(message)
            else:
                message = {"msg": "操作失败", "succe
ed": False, "success": False}
                return json.dumps(message)
```

首先，需要引入 web.py 模块，然后需要把 URL 结构告诉 web.py。该结构一般是一个正则表达式和接

受请求类的名称组合而成的。针对上述代码中的 '/manager/updateList?url' 请求，编码出 updateList 类。针对 post 请求，对传入的数据进行处理之后，调用数据库接口进行数据更新，最后返回处理结果。

5. 结语

本文的研究对象是：饲草料机械设备数据共享的应用需求，首先确定了数据共享的理论研究方向，包括饲草料机械设备数据的处理和存储。创新之处在于将 NoSQL 数据库 MongoDB 代替传统关系型数据在数据共享平台中的应用，利用 MongoDB 无“关系”的特点，提高数据共享平台数据录入以及查询的效率，同时在研究了机械设备数据的特点的基础上，应用 Python 语言的 web.py 框架和非关系型数据库 MongoDB，解决了饲草料机械设备数据格式复杂的问题，实现机械设备的发布和交互式使用，并可以有效整合饲草料行业相关机械数据，政府通过平台对饲草料生产机械发展状况予以了解和补贴，企业能够把自己生产的饲草料生产机械通过平台展销，科研院所的工作人员能够把新研制的农业装备让农户、企业了解，消费者(农户)也能够对自己的机具的配置提供信息支撑，是促进饲草料生产机械设备企业加快发展，更有效的推进饲草机械化推广的重要工具。

参考文献 (References)

- [1] 李娟, 刘德江, 江洪. 国际科学数据共享现状研究[J]. 图书馆建设, 2009, 2: 19-25.
- [2] 潘凡. 从 MySQL 到 MongoDB——视觉中国的 NoSQL 之路[J].程序员, 2010, 6: 79-81.
- [3] 黄贤立. NoSQL 非关系型数据的发展及应用初探[J]. 福建电脑, 2010, 7(30): 45.
- [4] 范凯. NoSQL 数据库综述[J]. 程序员, 2010, 6: 76-78.
- [5] K. Chodorow, M. Dirolf. MongoDB 权威指南. O'Reilly, 2010.
- [6] R. Mondéjar, P. García-López, C. Pairot and L. Pamies-Juarez. A transparent infrastructure for decentralized web deployment using distributed interception. Original Research Article Future Generation Computer Systems, 2013, 29(1): 370-380.
- [7] A. Martelli 等, 高铁军译. Python Cookbook. 北京: 人民邮电出版社, 2010.
- [8] H. Hackl, G. Stocker, P. Charoentong, B. Mlecnik, G. Bindea, J. Galon and Z. Trajanoski. Information technology solutions for integration of biomolecular and clinical data in the identification of new cancer biomarkers and targets for therapy. Review Article Pharmacology & Therapeutics, 2010, 128(3): 488-498.