

基于MEAN框架的实时数据共享竞赛系统的设计与实现

石 清

(武汉体育学院体育工程与信息技术学院, 湖北 武汉 430079)

摘 要: 在“互联网+”概念的影响下,越来越多的信息技术应用于体育产业。如何充分地利用信息技术手段为体育竞赛服务,如何使体育竞赛组织工作更加高效,如何使竞赛数据更加透明,是目前面临的重要问题。结合摩托艇项目,利用MEAN框架,开发了一套实时数据共享竞赛管理系统,并将基于RESTful接口的异步I/O模式贯穿于整个体育竞赛系统中。该系统能有效降低获取信息的成本,使参与者即时掌握赛事动态,提高工作效率,改善用户体验。

关键词: 实时数据共享;竞赛系统;MEAN框架;RESTful接口

中图分类号: TP311 **文献标识码:** A

Design and Implementation of the *Real-Time* Data Sharing Competition System Based on MEAN Framework

SHI Qing

(School of Sports Engineering and Information Technology, Wuhan Sports University, Wuhan 430079, China)

Abstract: With the influence of Internet+, more and more information technology is applied to the sports industry. The current major problems lie in how to make full use of information technology to serve sports competitions, how to make sports competition organization work more efficiently and how to make competition data more transparent. In conjunction with the motorboat project, the real-time data sharing competition system is developed based on the MEAN framework, applying the RESTful-based asynchronous I/O mode to the entire sports competition system. The system can effectively reduce the cost of obtaining information and update participants with latest news and events, increasing work efficiency and improving user experience.

Keywords: real-time data sharing; competition system; MEAN framework; RESTful interface

1 引言(Introduction)

近年来随着科技的进步,尤其是“互联网+体育”概念的提出,为体育产业带来了颠覆性的影响,为整个行业的发展提供了新模式。社交媒体的广泛应用不断推动民众参与体育的热情,无论是手机还是各种可穿戴设备都被广泛应用于体育相关的健身活动中。互联网视频平台及新媒体已经成为社会主流观赛和赛后讨论渠道,各式各样的体育信息平台如雨后春笋般涌现出来。无论是体育电商、赛事直播,还是个人健身、互动交流,各类应用满足了大众个性化体育信息服务的需求,从个人到企业组织,乃至整个社会的信息化程度越来越高。

信息化的过程就是数字化的过程,进而数据化的过程。但是,如果要发挥数据的真正价值,仅仅把数据储存在数据库中是不够的,而是要将数据进行分享。正如阿里巴巴首席

技术官王坚在其著作《在线》中所提到的,“今天数据的意义并不在于有多大,真正有意义的是让数据变得在线,这恰恰是互联网的特点。为什么今天的淘宝数据值钱,因为它是在线的。写在磁带和纸上的数据,作用是有限的^[1]。”随着云数据库技术的发展与运用,无论是哪种平台在运行的过程中产生的数据都可以存储在云数据库中,而这些海量的数据都可以通过接口的形式实现共享,为今后大数据的分析和挖掘提供了前提。大数据与云计算技术的发展也给体育服务带来了新的发展契机^[2]。通过数据共享,分析和挖掘数据背后的逻辑联系,进而有利于更加全方位的展现体育的轮廓和视角,无论是将这些分析结果应用于政策制定还是媒体传播,都能使其信息更具深度性^[3]。

随着我国体育赛事的逐渐增多,对竞赛服务体系也提出了更高的要求^[3]。而传统竞赛服务系统多采用C/S构架,前期

需要较大的硬件成本投入,同时开发周期长,成本高,除个别大型综合性赛事外,单项体育比赛的竞赛系统几乎不具备实时分享数据的能力。而普通基于B/S构架下的体育竞赛信息管理系统功能多集中在运动员注册与赛事报名上,功能较为单一。而一套基于B/S架构的轻量级全功能竞赛管理系统,无论是在数据分享,系统可扩展性上,还是在快速开发和维护成本方便都将具有更大的优势。

2 相关技术介绍(Related technology introduction)

MEAN框架是一个JavaScript平台下现代Web开发框架的总称,是MongoDB、Express、Angular、NodeJS四个框架的第一个字母组合的简称,该框架均使用JavaScript作为开发语言(Angular 2.0之后使用的是TypeScript),开发者无须在不同的语言之间切换,很好地适应了个人全栈开发的需求。

2.1 NodeJS介绍

NodeJS是一种基于Chrome V8引擎,用于搭建响应速度快、易于扩展的网络应用的执行平台。Chrome V8引擎针对JavaScript的编译进行了优化,使得代码运行速度获得了极大提升,同时节约了开发成本^[4]。

2.2 Express介绍

Express是一个基于NodeJS平台的极简、灵活的Web应用开发框架,它提供一系列强大的特性,可以方便地创建各种Web和移动设备应用。丰富的HTTP快捷方法和任意排列组合的Connect中间件,使创建健壮、友好的API变得既快速又简单^[5,6]。

2.3 Angular 介绍

Angular是一个基于HTML和TypeScript构建客户端应用的平台与框架。它是最著名的SPA框架之一,它能提供一种接近桌面应用程序的流畅用户体验单Web页面应用程序,它集声明式模板、依赖注入、端到端工具和一些最佳实践于一身,能轻松地构建Web应用。

2.4 MongoDB介绍

MongoDB属于NoSQL,其数据保存为JSON格式文档,具有高性能、高可用性和自动扩展功能。其支持的数据结构非常松散,可以存储比较复杂的数据类型。尤其适合项目早期开发过程中,数据结构尚未完全明确,需要高度灵活性的IT系统,并对数据库性能要求较高,但不需要高度的数据一致性的开发场景。

3 需求分析(Requirements analysis)

实时数据共享竞赛系统的设计与开发,是为了方便赛事的组织和管理,是通过互联网实现信息的实时交互,打通所有沟通环节,无论是运动员的报名报项,还是竞赛委员会公布比赛成绩,都完全通过竞赛系统完成。即实现了无纸化办公,又提高了工作效率。其具体的功能需求如下所述。

3.1 用户信息管理

面向的对象为运动员和俱乐部管理人员,主要功能是人及俱乐部基础信息管理,报名参赛信息管理等。

3.2 竞赛项目管理

面向的对象为赛事主办单位,主要功能是分布竞赛通

知、竞赛项目设定、日程安排等赛事相关信息。

3.3 竞赛信息管理

面向的对象为裁判委员会,主要功能是调整出发艇位,为竞速类项目计时计圈,为表演类项目打分,对犯规进行处罚,计算项目积分,显示实时成绩,分布最终成绩等。

4 系统架构(System architecture)

4.1 服务器架构

为了提高系统安全性,以及后续的扩展能力,设计时采用将计算、数据库和文件存储分离的架构。整个系统都部署在阿里云体系中,即兼顾了高效率的弹性又缩减了自建服务器的成本,服务器构架如图1所示。

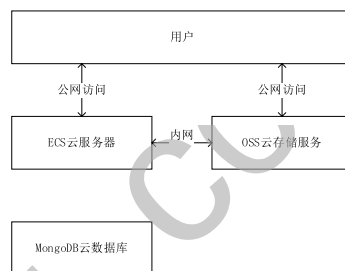


图1 服务器架构模型

Fig.1 Server architecture model

4.2 MEAN架构

基于NodeJS、Express和MongoDB的架构可以方便的在服务器端创建一个MVC框架的系统。MongoDB结合Mongoose可以很好地管理模型层并进行持久化,所有的业务逻辑通过NodeJS和Express来完成^[7-9]。服务器端无须设计View层,所有的数据都采用RESTful接口的形式向客户端传输数据,所有接口都是通过Express路由route来设计完成。Angular通过Ajax向上述接口请求数据,并将获取的数据绑定在Model对象上,并最终在View层中显示出来^[10-12]。

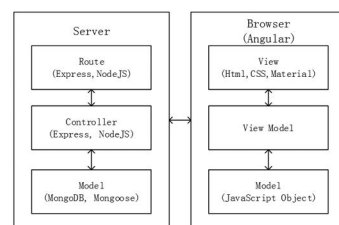


图2 MEAN框架结构模型

Fig.2 MEAN frame structure model

对于一个用于解决体育竞赛的Web应用程序来说,以轻量级去适应体育竞赛本身的灵活多变,通过上述多层架构实现一个可重用性、简单性、可扩展性、功能清晰分离的系统。

5 系统的设计与实现(System design and implementation)

5.1 NoSQL数据库的必要性

作为体育竞赛系统来说,其系统结构和数据结构受制于体育项目的竞赛规则和竞赛组织形式。而竞赛规则间隔几年就会有一次调整,竞赛形式也会根据具体竞赛条件进行调整。尤其是小众的运动项目,变化就更为明显。其根源上就已经决定了很难在事务处理上保持数据结构的恒定不变。而

NoSQL由于其本身数据模式灵活, 适合非结构化的数据存储。

在使用MongoDB来存取数据时, 通常都是使用Schema来定义对象。它无须定义MySQL等关系型数据库的id主键, 系统会自动为每个文档添加一个"_id"键, 其默认类型为ObjectId对象, 该对象并不属于自增型数据, 当有大量数据同时进行请求时, 它无须等待上一个id序号存储完成之后再行存储。对于体育竞赛中多场次同时竞赛的方式具有非常好的支持, MongoDB的该特性非常适合于异步IO的存储方式。

5.2 数据模型的设计与实现

使用NoSQL数据库MongoDB来存取数据, 与使用其他关系数据库(如MySQL)的语法区别并不明显, 转换成本不高。而对于一个数据结构尚处于变化中的系统来说, 使用MongoDB可能还具备一定的优势。MongoDB本身提供的查询语言非常强大, 同时配合Mongoose, 以类似于面向对象的方式来操作数据对象, 实现数据的持久化。

阿里云出于完全考虑, 其MongoDB数据库通常只能通过内网进行访问, 所以在开发过程中不能实时连接云数据库进行调试, 只能在本地建立一个相同的MongoDB数据库。为了开发方便建立一个独立的database configuration文件, 单独保存数据库的设置, 分别用于网络与本地的数据库连接。

在NodeJS的app.js文件中配置MongoDB数据库, 分别在本地和阿里云数据库建立相同的数据库用于前期测试:

连接本地MongoDB数据库:

```
module.exports={
  databaseUrl:'mongodb://localhost:27017/
motor',
}
连接阿里云MongoDB数据库:
module.exports={
  databaseUrl:'mongodb://dbadmin:*****@
ddsdp103d3f778256241.mongodb.rds.aliyuncs.com:3717/
motor',
}
```

在生成数据模型时, 主要是应用mongoose的Schema来定义数据库的结构。每个Schema会映射到MongoDB中的一个collection, 使用mongoose的model()方法将Schema编译为Model, 从而实现Model层的创建。

5.3 文件管理服务的设计与实现

系统在设计时采用将WEB服务器与文件服务器分离的模式, 这样可以分担Web服务器的I/O负载, 提高WEB服务器的性能和稳定性。同时, 将耗费资源的文件服务分离出来, 也可以有效地减少带宽成本, 提高访问速度。

创建文件服务器时选择的是阿里云的对象存储服务(Object Storage Service, OSS)。OSS拥有与平台无关的RESTful API接口和SDK开发包, 提供包括图片格式转换、缩略图、剪裁、水印、缩放等多种增值服务, 结合OSS进行开发, 将使整个开发过程方便、快捷, 并使系统的文件管理服务功能更加强大。

系统中的文件管理服务开发是将OSS封装在接口中, 从而实现对接口的调用。封闭图片上传的函数定义如下:

```
const OSS=require('ali-oss');
const client=new OSS({
  region: 'oss-cn-hangzhou',
  accessKeyId: 'AccessKeyId',
  accessKeySecret: 'AccessKeySecret',
  bucket: 'motor'
});
```

一般来说, 将文件上传到分离的WEB服务器和文件服务器上时有两种方式。第一种是先把文件上传到WEB服务器上, 然后再转到文件服务器; 第二种是直接上传到文件服务器上。系统在开发过程中选择的是第一种模式, 这主要是出于安全考虑, 将OSS的写入和删除等功能设置为只能通过内网ECS才允许调用的模式, 外网只允许对OSS中的文件进行读取。同时OSS对于内网传输实行免费的计费模式也同时保证了该方案的经济性。所以在项目设计和实施的时, 所有文件在上传过程中都必须从ECS服务器上进行中转, 中转完成后原文件将在ECS服务器上立即删除。如图3所示。

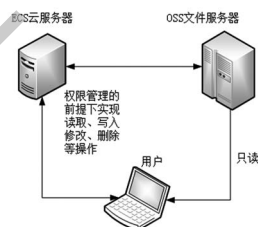


图3 OSS文件服务器交互模型

Fig.3 OSS file server interaction model

创建的RESTful API接口调用上述函数, 完成对整个图片上传服务的封装。系统中其他任何模块需要上传图片功能时, 仅需要调用该API接口, 即可返回图片上传成功后的地址。

5.4 RESTful API接口的设计与实现

该体育竞赛系统建立在B/S构架之上, 它不仅需要满足竞赛团队在浏览器端发送的请求, 同时还需要兼顾为比赛现场大屏幕提供实时竞赛数据, 以及为电视台转播提供实时竞赛数据的需求。所以, 系统的应用场景是一个IO密集、高并发的应用场景。而Node的异步I/O可以让JavaScript的单线程很好的处理阻塞问题, 使系统更好地利用云服务器的计算资源。同时, 体育赛事的竞赛数据本身需要其数据具备高可靠性, 而Node异步I/O、非阻塞的特性正好满足体育比赛的这些特征。一个基于Node框架下的RESTful API接口的WEB软件架构不仅可以完美支持上述需求, 还可以应对未来其他系统或其他终端对数据资源的需求。

系统的业务逻辑和流程控制都是由独立的Controller对象来实现, 针对具体的业务流程, 会有不同的控制器。以下仅以Message对象简要说明实现过程。Message Controller对象定义如下:

```
var Message=require('../models/message');
exports.getAllMessages=function(req,res) {
```

```

    Message.find({}).sort('created').exec(function
(err, collection) {
    if(err) {
        res.status(500).json({msg:err});
    } else {
        res.status(200).json({collection});
    }
    })
};

```

在定义完Controller对象之后,将通过Express的Route路由来实现对该资源的调用。在NodeJS中创建一个独立的api.js文件用于存储所有的API接口信息,所有对该路由文件的访问地址就变成了https://example.com/api/**。然后再通过Get、Post、Put、Delete等方式对资源进行操作。Message资源的路由地址定义如下:

```

const message_ctrl=require('../controller/message.
controller');
router.get('/all-messages',message_ctrl.
getAllMessages);

```

5.5 RESTful API接口的权限管理

对于一个设计好的RESTful API接口来说,其本身是无状态的,用户可以从任意一种可以发送HTTP请求的终端向服务器发送请求。虽然这样做保证数据的高效分享,但却存在极大的安全隐患。系统在设计的过程中,将所有需要身份验证和角色权限控制的接口只用node中间件的方式进行管理。使用JSON Web Token来验证请求是否为注册用户发送,通过验证注册用户role字段中是否包含特定数据,判断用户是否具有操作权限。

5.6 前端界面的设计与实现

Angular作为Google公司主推的前端应用框架,其自动化双向数据绑定技术使Angular运行操作体验与本地应用非常类似。而模块化设计和依赖注入等技术都大大地提高了开发速度和软件结构的弹性。系统前端界面的设计过程中充分利用Angular的特点进行设计,充分考虑用户体验,无论是用户输入还是查询都具备Autocomplete功能。考虑到该竞赛系统主要应用于国际赛事,使用Angular第三方组件ngx-translate完成多语言支持。通过独立编辑en.json文件和zh.json文件完成语言包的开发,也可根据需要开发对更多种类语言的支持。语言包中所有定义的变量都将通过Angular pipe的方式最终在页面中呈现,应用方法如下:

```
<div>{{ 'Event_title' | translate }}</div>
```

最终实现了一个能够进行中英文切换,并满足摩托艇国际大赛需求的竞赛服务系统。无论是赛事主办单位、裁判员、运动员,还是现场观众都可以通过该系统实时了解赛事资讯。

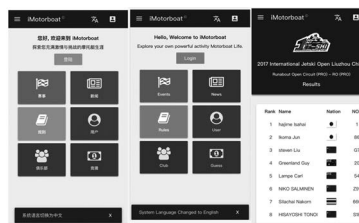


图4 软件最终界面截图

Fig.4 Software final interface

6 结论(Conclusion)

本文基于MEAN框架开发的摩托艇竞赛系统,利用将计算、数据库和文件存储分离的架构提升了服务器端的安全性,结合RESTful API接口的设计提升了服务器端的易用性和可扩展性。在系统使用过程中,通过线上线下相结合的模式,大幅降低赛事参与各方的沟通成本,有效提高办赛效率,提升所有赛事参与者的用户体验。该系统已成功应用于2017年中国柳州国际水上摩托公开赛,并首次在摩托艇项目中实现了无纸化办赛,即节约了办赛成本,也让所有赛事参与者享受到“互联网+体育”带来的便利。

参考文献(References)

- [1] 王坚.在线[M].北京:中信出版社,2018.
- [2] 王朝霞.互联网环境下体育产业服务发展模式创新研究[J].广州体育学院学报,2016,36(5):18-21.
- [3] 周兵.大数据时代新闻报道中数据新闻的运用[J].中国统计,2018(8):20-22.
- [4] 陶倩.基于竞赛规程的大型体育赛事竞赛成绩管理软件系统模型研究[J].山东体育科技,2014(1):62-66.
- [5] 黄扬子.基于NodeJS平台搭建REST风格Web服务[J].无线互联科技,2015(16):57-59.
- [6] 王伶俐.基于NodeJS+Express框架的轻应用定制平台的设计与实现[J].计算机科学,2017(11):596-599.
- [7] Haviv,Amos Q.MEAN Web Development:Master real-time web application development using a mean combination of MongoDB,Express,Angular JS,and Node.js[M/OL].Birmingham,UK:Packet Publishing.
- [8] Holmes,Simon(2015).Getting MEAN with Mongo,Express,Angular,and Node[M/OL].Shelter Island,NY:Manning Publishing.
- [9] Dickey,Jeff(2014).Write Modern Web Apps with the MEAN Stack:Mongo,Express,AngularJS, and Node.js(Develop and Design)[M/OL].San Francisco,CA:Peachpit Press.
- [10] 程桂花,沈炜,何松林,等.Node.js中Express框架路由机制的研究[J].工业控制计算机,2016,29(8):101-102.
- [11] 朱建兵.基于Node.js高并发网络应用框架的研究与实现[D].北京:北京邮电大学,2013.
- [12] 周昆.基于Node.js的非阻塞多核心服务器设计与实现[D].厦门:厦门大学,2011.

作者简介:

石清(1981-),男,硕士,讲师.研究领域:体育信息技术,人机工程学.