

应用技术

## Node.js 及 MongoDB 应用研究

李 鹏<sup>1,2</sup>

(1. 英业达集团(天津)电子有限公司 天津 300193; 2. 天津科技大学电子信息与自动化学院 天津 300457)

**摘 要:** 基于 Node.js 技术对客户关系管理系统进行搭建, 设计出丰富的库函数和助手函数, 以实现开发与测试的高效率。新构建的可供电商企业使用的客户关系管理系统(Customer Relationship Management System, CRM)能够实现对客户信息的集中管理和共享利用, 即客户资源的企业化管理, 避免因业务调整或人员变动造成的客户资源流失和客户管理盲区的产生; 更重要是, CRM 可以基于完善的客户信息支持不同业务面向客户的工作, 实现客户信息在企业内部的充分共享利用, 提高面向客户工作的有效性和效率, 从而全面提升客户的满意度。

**关键词:** 服务器管理软件 Web 服务 软件开发

中图分类号: TP311.1

文献标志码: A

文章编号: 1006-8945(2015)06-0034-03

DOI:10.14099/j.cnki.tjkj.2015.06.013

### Applied Research of Node.js and MongoDB

LI Peng<sup>1, 2</sup>

(1. Inventec (Tianjin) Electronic Co., Ltd., Tianjin 300193, China; 2. College of Electronic Information and Automation, Tianjin University of Science & Technology, Tianjin 300457, China)

**Abstract:** Based on Node.js, a Customer Relationship Management System software was built with abundant library functions and helper functions to achieve high efficiency in the developing and testing. Then, it was used to set up a Customer Relationship Management (CRM) System for electric commercial enterprises, realizing the centralized management and sharing of customer information, namely, enterprise-style management of customer resources, to avoid the loss of customer resources and customer management blind spots generated due to changes resulting from business restructuring or personnel alteration. More importantly, the CRM, based on improved customer information, is able to support different customer-facing works to achieve full customer information sharing within the enterprise, improve the effectiveness and efficiency of customer-oriented tasks, therefore improving the overall customer satisfaction.

**Key words:** enterprise server management software; Web service; software development

目前, 国内外市场上的客户关系管理系统(CRM)多采用 Java 作为基础语言进行设计开发, 其安全性、稳定性、系统完善度、架构安全性等得到了很好的保证。但是其开发周期相对较长, 人力需求较多, 最重要的是 Java 在高并发的处理上存在一些弊端, 很容易造成 Server 端阻塞。而 Node.js 高效的异步事件调度机制和基于 javascript 的闭包模式较好地解决了这些弊端。

### 1 Node.js 技术及应用概述

#### 1.1 Node.js 的定义、功能

Node.js 是由 Ryan Dahl 编写的服务器端 javascript, 其初衷是为了编写更为高效的 Web 服务器, 它使用当前最快的谷歌 V8 javascript 引擎和单线程的模式。因为不需要考虑并发, 因此也就不涉及锁和阻塞的问题, 大大简化了编程。它的事件回调模型均采用异步操作, 如数据库访问都是通过事件来触发的。

Node.js 完全发挥了 javascript 作为动态解释语言的强大威力, 开发人员可以自由使用 javascript 的一切特性, 如闭包(closure), 并且不需要担心跨浏览器支持(因为是服务端)。那么单线程如何处理多用户请求呢? 事实上, Ryan 观察到 Web

访问的一个事实: 每次 Web 请求服务周期最耗时间的往往是 I/O 操作, 包括读写文件、数据库操作、开启网络等。真正 Web 服务内部的运算只占很小的比例。例如浏览器访问一个页面, 几乎大部分时间都花在 I/O 的读写上, 从读文件一直到向接口(socket)写数据, 花在页面后端代码的计算量则微乎其微。举例来说, 假设 Web Server 是一个银行, 每个浏览器的访问相当于客户去窗口取一笔钱, 客户在某个窗口办理业务相当于服务器从本地硬盘读写文件, 这个操作会花很长时间, 由于窗口数有限, 所以需要排队叫号系统, 即 Node.js 线程。当窗口有空时, 营业员就通知排号系统叫下一位, 但是喇叭一次只播放一个喇叭的声音, 所以所有通知都存放在了一个事件队列中, 排号系统报完一个号再处理下一个, 直到队列空为止, 然后就处于空闲状态。如果把叫号类比后端代码执行, 它的速度就会比处理银行业务快得多, 所以只需一个喇叭就足够了。由于没有多线程的切换消耗, 所有运算都作用在实际的计算上, 从而提高了处理量。

#### 1.2 Node.js 的优势

Node.js 在实时的 Web 应用上采用了基于 WebSocket 的推送技术。这意味着在经过了 20 多年的基于无状态的请求-返回机制的无状态交互之后, 终于有了实时的、双向连接的

收稿日期: 2015-05-07

Web 应用,客户端和服务端都可以发起通信,能够自由地交换数据。与此形成鲜明对比的是传统的 Web 响应模式,客户端总是主动发起通信而服务端被动返回。此外,这些都是基于运行在标准 80 端口上的开放 Web 组件(HTML、CSS 和 JS)。

尽管 Flash 和 Java Applet 的形式已经应用很多年了,但实际上这些方式只是使用网络将数据传递到客户端上的沙箱(Sandboxie)环境。它们都是隔离运行的,而且经常操作到需要额外的权限之类的非标准端口。凭借独特的优势,目前 Node.js 已在许多著名公司的产品中发挥了关键作用。

### 1.3 Node.js 的工作原理

Node.js 的主要思路是:使用非阻塞的、事件驱动的 I/O 操作来保持在处理跨平台(across distributed devices)数据密集型实时应用时的轻巧高效。简单讲,Node.js 不是一个即将主导 Web 开发的银弹(silver bullet)级的平台。相反,它是一个满足特别需求的平台。事实上,使用它进行繁重的计算等于摒弃 Node 所有的优点。Node 真正的亮点在于建设高性能、高扩展性的互联网应用,因为它能够处理庞大的、高吞吐量的并发连接。

传统的网络服务技术,是每新增一个连接(请求)便生成一个新的线程,这个新的线程会占用系统内存,最终会占据所有的可用内存。而 Node.js 使用非阻塞的异步 I/O 调用,仅运行在一个单线程中,所有连接都由该线程处理,全部挂在该线程的事件循环中,可以允许其支持数万个并发连接。

当所有客户端请求共享单一线程时 Node.js 也会出现问题,这也是编写 Node.js 应用的潜在缺陷。首先,大量的计算可能会使 Node 的单线程暂时失去反应,并导致所有的其他客户端的请求一直阻塞至计算结束。其次,开发人员需要非常小心地避免 Exception 阻塞核心的事件循环,因为这将导致 Node.js 实例的终止(即程序崩溃)。

## 2 CRM 软件结构及数据库介绍

### 2.1 CRM 管理系统的软件架构

CRM 管理系统的软件架构组成如图 1 所示:

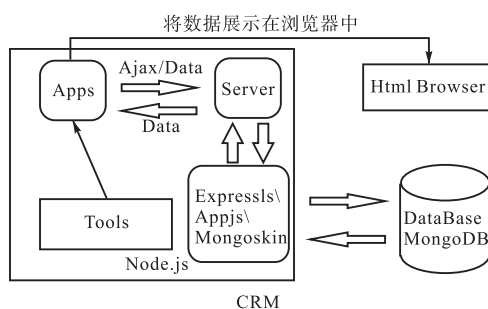


图 1 CRM 管理系统的软件架构

Fig.1 Software architecture of CRM management system

图 1 显示了 CRM 系统的工作流程:浏览器端通过 Apps 发送请求(Ajax)至后台(Server),后台寻找此请求所对应的方法,这些方法通过与数据库的交互工具(MongoSkin 等)来实现对数据的增、删、改、查,再把相应的结果返给 Apps,然后通过 Html 及 javascript 显示在浏览器(Browser)段,这样就可以将客户数据展示在浏览器端对应的功能区,也可以实现对数据库中所有用户数据的增、删、改、查等处理。

CRM 系统中的网页主要是基于 HTML、CSS 和 javascript 技术实现的,再通过 JQuery、Bootstrap 等辅助工具来更好地实现所需功能和页面的美观。

### 2.2 CRM 系统所用数据库 MongoDB 介绍

开发一个 Web 应用程序,最重要的莫过于数据库的使用。过去 PHP 有 MySQL 当最佳伙伴,而现在 Node.js 有 MongoDB 做最佳的组合。MongoDB 是 NoSQL(非关系型数据库)的代表之一,它是一个基于分布式文件存储的数据库,由 C++ 语言编写,其采用 JSON/BSON 作为数据储存和沟通的格式,亦使用 javascript 做为服务器端(Server-side)的执行语言,一切设计习惯都与 Node.js 非常吻合,搭配使用也方便快捷。

该系统使用 MongoDB 数据库,它拥有 NoSQL 的普遍特性,不用预先定义数据结构(Schema),并且能够胜任大数据量下任意字段的查询,写入性能高,可写入百万级别的数据。MongoDB 比普通关系型数据库快。因此,针对 CRM 系统大数据量的用户数据和 ESMS 架构所采用的 Node.js 技术,使用 MongoDB 数据库。

## 3 CRM 管理系统功能模块设计

CRM 管理系统含用户登录、系统首页平台、用户管理、商品分析等功能模块,基于 Node.js、Html、javascript 等技术应用 MongoDB 数据库搭建了些功能平台,用户可以依据兴趣或需求点击相关按钮进入相关页面查看。

### 3.1 用户登录界面模块

用户使用账号、密码完成登录,登录成功会看到系统首页,如登录失败会有失败提示。登录界面如图 2 所示。



图 2 登录界面  
Fig.2 Login interface

### 3.2 系统界面设计

系统首页分为 3 部分内容:①标题栏为业务功能区,通过点击标题栏中的功能区可以进入到相对应的功能模块;②报表分析区,通过采样将客户数据进行分析 and 整理,然后以图形的方式直观地显示出来;③登入用户显示及注销区,用户点击登录名后的下拉箭头,可以选择退出系统,也可以修改个人信息情况。

系统首页界面如图 3 所示。

### 3.3 客户管理模块设计

客户管理模块界面如图 4 所示。

该模块主要有:会员列表、会员自定义分组、会员等级设定等功能。用户点选菜单中的 [客户管理] → [会员列表] 功能,系统显示会员列表界面。会员列表界面可以支持多种分类查询,查询功能包含分组查询和条件查询。

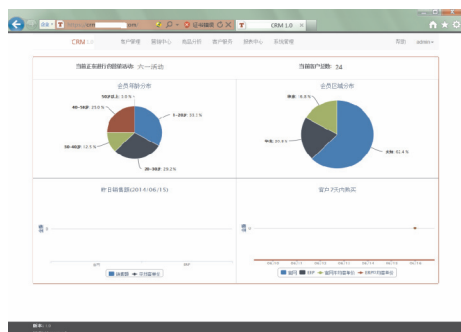


图 3 系统首页界面

Fig.3 Interface of system homepage



图 4 客户管理模块界面

Fig.4 Interface of client management module

在页面中输入汇入渠道、姓名、性别、手机号等字段中任意要搜索的值,点击确定,触发以下方法并传入相应参数:

params: {'name': {'\$like': '页面输入的值'}}

```
function getUserList (params) {
  var bizUrl = '/customers/memberlist/';
  $.ajax ({
    type: 'GET'
  , url: bizUrl
  , data: {'param': JSON.stringify (params) }
  , dataType: 'json'
  , success: function (json)
  {
    CreateMemberTable (json.data) ;
  }
  });
}
```

由上述方法将参数输入 server 端的 crm/routes/crm/rpc/customer/member.js 文件,通过下述接口:

```
}else if (json['path'] = '/ memberlist /') {
  getMemberList (JSON.parse (json['get'])['param']) , json ,
function (ret) {
  res.end (JSON.stringify (ret['result']));
});
}
找到对应的后台方法 getMemberList ();
function getMemberList (param, json, fn) {
  var tableName = 'user.local';
  var collection = _db.collection (tableName) ;
  collection.find (param) .toArray (function (err, result1) {
    if (err == null)
    {
```

```
json['result']['data'] = result1 ;
} else {
  json['result']['data'] = false ;
  json['result']['errcode'] = 1 ;
  json['result']['errmsg'] = JSON.stringify (err) ;
}
fn (json) ;
});
}
```

最后将查找出的 json 数据返回前台,前台会据此画出对应的表格等操作。

通过以上前后台数据传递代码可以看出,当请求发到 Node.js 后,它会采用回掉(CallBack)的方式来处理事件,整个过程没有阻塞也不需要等待。基于这样的机制,理论上陆续有用户请求连接,Node.js 都可以进行响应,因此 Node.js 能支持比 Java、PHP 程序更高的并发量,虽然 Java、PHP 也可以通过子线程的方式来实现并发请求,但显然 Node.js 的回掉函数和异步机制性能更高。由于要实现的操作系统可能会有很多的并发情况和 I/O 请求,因此采用 Node.js 和 MongoDB 的开发模式会更加高效易用。

同样,其他的一些功能模块也是采用类似的逻辑,只是按照不同的需求而搭建不同的页面结构来完成,如商品分析、客户服务、系统数据管理等功能。

## 4 结 语

介绍了 Node.js 以及以 Node.js 为基础开发框架的应用系统 CRM。它使用 Google 的 V8 虚拟机来解析和执行 javascript 代码,也可以说 Node.js 就等于“运行时环境+库”。它使用单线程和非阻塞 I/O,这使得架构性能更好,开发代码更为整洁,同时 Node.js 是一个非常靠近底层的工具库或运行环境,可以精细控制 Request 和 Response 的时间和内容,而且它的异步回调模式能够更好地处理大并发事件。因此,CRM 管理系统使用 Node.js 可以更好地处理多人操作的并发情况以及大数据量所带来的问题,通过 Node.js 的异步模式和其特有的监听方式,可以更好地对用户数据进行统计和采样,使操作人员更好地掌握用户的相关数据。

随着信息化服务时代的到来,Web 服务的高效性和准确性也日益重要。综合 Node.js 的优势,其在未来的 Web 开发和其他软件开发领域一定会得到广泛应用■。

## 参考文献

- [1] 李松峰. JQuery 基础教程[M]. 北京: 人民邮电出版社, 2012.
- [2] [美]Mike, Wilson 著; 林冀, 范俊, 张鹏译. Node 应用程序构建[M]. 北京: 人民邮电出版社, 2014.
- [3] 黄丹华. Node.js 开发实战详解[M]. 北京: 清华大学出版社, 2014.
- [4] 郭远威. 大数据存储 MongoDB 实战指南[M]. 北京: 人民邮电出版社, 2015.

下转第 39 页

水收集利用技术发展的过程中,国家也不断出台相应的政策对雨水收集利用提供支持,制定出法律法规,使雨水回收利用有章可循、有法可依。■

#### 参考文献

- [1] 于宵. 城市道路路面雨水收集与利用系统设计[J]. 城市道桥与防洪, 2013(6): 110-114.
- [2] 刘滨谊, 许珊. 利用雨水收集系统解决景观用水问题[J]. 中国园林, 2007, 23(2): 54-56.
- [3] Domènech L, Saurí D. A comparative appraisal of the use of rainwater harvesting in single and multi-family buildings of the Metropolitan Area of Barcelona(Spain): social experience, drinking water savings and economic costs[J]. *Journal of Cleaner Production*, 2011, 19(6): 598-608.
- [4] 梁文逵. 城市雨水收集利用研究现状与进展[J]. 工业用水与废水, 2014, 45(3): 6-9.
- [5] Rahman A, Keane J, Imteaz M A. Rainwater harvesting in Greater Sydney: water savings, reliability and economic benefits[J]. *Resources, Conservation and Recycling*, 2012(61): 16-21.
- [6] 常素云, 孙井梅, 郑毅, 等. 新型居住区良性水循环工程设计[J]. 中国给水排水, 2009, 25(6): 56-60.
- [7] 朱平, 王维平, 曹彬, 等. 城市屋面雨水收集利用工程设计分析[J]. 地下水, 2012(6): 218-219.
- [8] 王晓玲, 张宝军, 白建国. 高校校园雨水收集利用研究现状探讨与分析[J]. 山西建筑, 2014, 40(28): 129-130.
- [9] 付尧涵. 雨水收集利用方法在公园设计中的应用——以北京大兴生药基地公园景观规划设计为例[J]. 工程建设与设计, 2013(1): 108-112.
- [10] 周云川, 张曦, 李显秋, 等. 雨水收集系统在工业建筑中的应用[J]. 四川建筑科学研究, 2014, 40(6): 228-230.
- [11] 脱宁, 胡强. 大型体育建筑开合式屋盖屋面雨水收集和排放[J]. 给水排水, 2006, 31(12): 65-67.
- [12] 甘灿, 万华. 城市雨水收集与利用研究[J]. 山西建筑, 2009, 35(7): 202-203.
- [13] 毕守海. 全国地下水超采区现状与治理对策[J]. 地下水, 2003, 25(2): 72-74.
- [14] 孙驹治. 南京: 全面实施雨水收集利用工程[J]. 建设科技, 2010(21): 52-55.
- [15] Palla A, Gnecco I, Lanza L G, et al. Performance analysis of domestic rainwater harvesting systems under various European climate zones[J]. *Resources, Conservation and Recycling*, 2012(62): 71-80.
- [16] Morales-Pinzón T, Lurueña R, Rieradevall J, et al. Financial feasibility and environmental analysis of potential rainwater harvesting systems: A case study in Spain[J]. *Resources, Conservation and Recycling*, 2012(69): 130-140.
- [17] Mahmoud W H, Elagib N A, Gaese H, et al. Rainfall conditions and rainwater harvesting potential in the urban area of Khartoum[J]. *Resources, Conservation and Recycling*, 2014(91): 89-99.

上接第 36 页

- [5] Bowers M, Synodinos D, Sumner V. Pro HTML5 and CSS3 Design Patterns[Z]. 2013: 1.
- [6] 李安瑜. Web Service 技术与实现[M]. 北京: 国防工业出版社, 2003.