

# 基于 MongoDB 的物联网接入云服务平台

何 倩<sup>1,2</sup>, 陈亦婷<sup>1,2</sup>, 董庆贺<sup>3</sup>, 李雄英<sup>1,2</sup>

- (1. 桂林电子科技大学 广西云计算与大数据协同创新中心, 广西 桂林 541004;  
2. 桂林电子科技大学 认知无线电与信息处理省部共建教育部重点实验室, 广西 桂林 541004;  
3. 桂林电子科技大学 广西自动检测技术与仪器重点实验室, 广西 桂林 541004)

**摘 要:** 为对海量智能设备的信息进行接入、存储、处理、挖掘其价值, 设计并实现了基于 MongoDB 的物联网接入云服务平台。该平台包括感知层、接入层、存储层、服务层和应用层, 利用分片的 MongoDB 和负载均衡技术, 能实时接入并存储海量设备数据, 通过 REST 架构实现灵活的系统组合, 根据不同设备和不同条件统计、分析设备数据, 按照地址、操作、检测值等挖掘数据信息, 并将结果以地图、统计图和统计表等直观方式展示。实验数据验证了该平台的有效性, 并达到了预期的运行效果。

**关键词:** 物联网接入; MongoDB; 云服务; REST 服务

中图分类号: TP393 文献标志码: A 文章编号: 1673-808X(2017)01-0001-07  
DOI:10.16725/j.cnki.cn45-1351/tn.2017.01.001

## A cloud service platform based on MongoDB for internet of things accessing

HE Qian<sup>1,2</sup>, CHEN Yiting<sup>1,2</sup>, DONG Qinghe<sup>3</sup>, LI Xiongying<sup>1,2</sup>

- (1. Guangxi Collaborative Innovation Center of Cloud Computing and Big Data,  
Guilin University of Electronic Technology, Guilin 541004, China;  
2. Key Laboratory of Cognitive Radio and Information Processing of the Ministry of Education,  
Guilin University of Electronic Technology, Guilin 541004, China;  
3. Guangxi Key Laboratory of Automatic Detection Technology and Instrument,  
Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** In order to access, store and process the massive information of the smart device and then dig its value efficiently, a cloud service platform based on MongoDB for internet of things accessing is designed and implemented. This platform consists of cognition, accessing, storage, service and application layers, provides high accessing and storage capability for the smart devices in real-time through the sharding MongoDB and load balance technologies, and realizes the flexible combination of system using REST architecture. These data can be statistically analyzed according to different devices and conditions, the valuable data information is mined based on address, operation and detection value and the results are presented as such visual form as map, statistical chart and statistical tables. The system architecture, implementation mechanisms and key technologies of this service platform are presented in detail. After deploying the accessing cloud on the Aliyun, the function and performance of the cloud service are evaluated.

**Key words:** internet of things accessing; MongoDB; cloud service; REST service

21 世纪以来, 物联网(internet of things, 简称 IoT)<sup>[1]</sup> 技术快速发展, 已应用到智能家居、远程医疗、环保、交通、农业、物流、公共安全、环境监测等领域。美国国家职能委员会预测, 到 2025 年物联网节点将存在于任何物体之内, 不仅包括各种电子产品, 还包括食品包装、家具、纸质文件等<sup>[2]</sup>。基于物联网

的智能物体(smart object)<sup>[3]</sup>的发展速度更是远远超出了大多数人的想象。数量如此多的智能设备, 将产生海量的操作、监控数据, 且具有巨大的价值。如何构建一个统一的信息处理平台, 有效地对海量智能设备数据进行接入、存储、处理、挖掘其价值, 成为一个亟待解决的问题。

收稿日期: 2016-12-24

基金项目: 国家自然科学基金(61661015); 广西云计算与大数据协同创新中心基金(YQ16801); 认知无线电与信息处理教育部重点实验室主任基金(CRKL160101); 广西自动检测技术与仪器重点实验室主任基金(YQ15102)

通信作者: 何倩(1979—), 男, 湖南安仁人, 教授, 博士, 研究方向为分布式计算、信息安全。E-mail: heqian@guet.edu.cn

引文格式: 何倩, 陈亦婷, 董庆贺, 等. 基于 MongoDB 的物联网接入云服务平台[J]. 桂林电子科技大学学报, 2017, 37(1): 1-7.

针对物联网数据接入服务,研究人员提出了多种实现方案。韩海雯等<sup>[4]</sup>提出采用云平台存储和处理设备数据,利用 SOA 架构实现面向服务的体系结构。Zhang 等<sup>[5]</sup>基于 DNS 提出一种物联网数据共享框架 IEDNS,但仅在理论上进行了探讨。由于 REST 服务架构采用 HTTP 协议,利用标准化动作,降低了开发难度。所以,目前物联网中的实体服务更多的是基于 REST 架构。文献[6-7]采用基于 REST 的软件架构,获取设备采集的数据,且用户可通过 REST 接口控制设备,实现物理设备和用户的交互。

物联网数据主要采用关系型数据库(SQL)和非关系型数据库(NoSQL)2 种方式进行存储。由于关系型数据库技术十分成熟,所以大多数的物联网数据都基于关系型数据库进行存储。Gonzalez 等<sup>[8]</sup>针对物流数据提出一种数据模型 RFID-CUBOID,将物流数据进行有效压缩以提高存储和查询效率。Yang 等<sup>[9]</sup>将采集的数据存储在 MySQL。丁治明等<sup>[10]</sup>基于 RDBMS 提出一种面向物联网海量传感器的数据库集群系统框架 IoT-ClusterDB,该框架利用一种双层树型结构,采用数据库集群的方式存储。纯粹基于关系型数据库技术难以满足物联网海量数据存储、实时查询和模糊查询等需求。

NoSQL 数据库不需要预定义模式、弹性可扩展、

高可用等特点,引入 NoSQL 数据库成为新的研究和应用热点。文献[11-12]提出基于 NoSQL 存储物联网数据。Francesco 等<sup>[13]</sup>提出一种基于 NoSQL 的海量物联网数据存储方案 IOTMDB,采用一种统一的方式进行表示和管理数据,但是对于不同种类的数据,如何选择对应的 NoSQL 数据库以及物联网数据时空关系研究不够。田野等<sup>[14]</sup>提出基于 NoSQL、REST 的物联网数据存储与共享策略。

鉴于此,结合智能设备数量大、种类多、用户对信息智能分析需求高的特点,设计并实现了基于 MongoDB 的智能物联接入云服务平台。平台采用 REST 架构及接口的方式实现物联网采集端与服务端、服务器端与应用端之间的交互,利用 MongoDB 集群存储物联网智能设备数据,满足海量数据存储和高效查询的需求,利用多种统计图、统计表、地图等可视化方法提供人性化的数据展示。

## 1 系统架构

基于 MongoDB 的智能物联云平台采用分层的体系架构,从下到上分别为感知层、接入层、存储层、服务层和应用层。整个平台的各个系统和数据库都部署在阿里云服务器上,可针对设备接入数量的多少进行动态扩展。系统架构如图 1 所示。

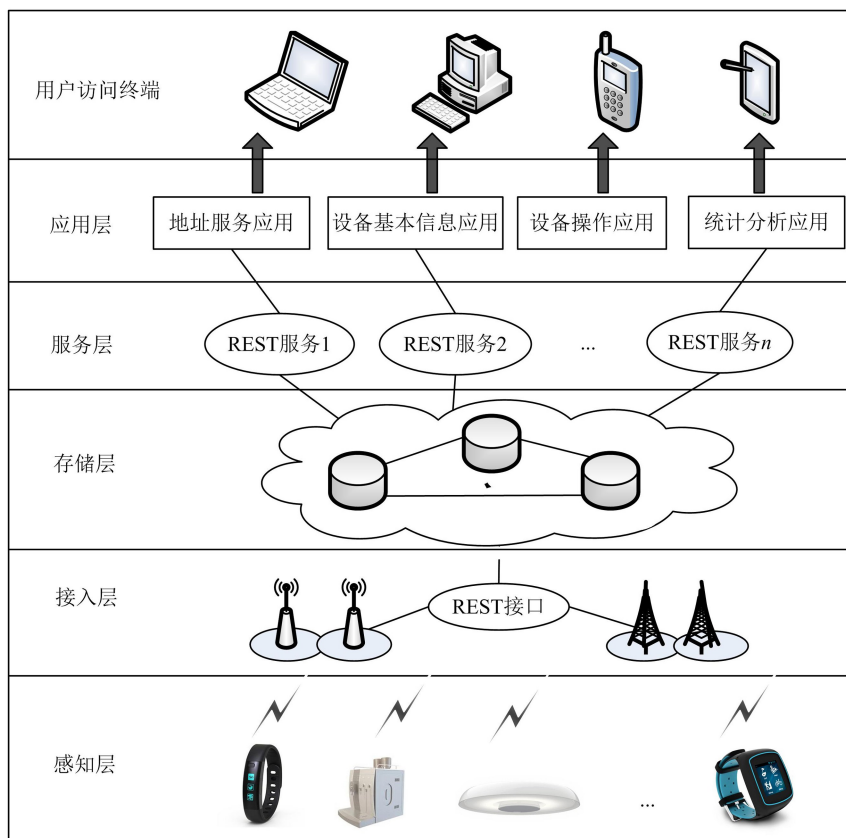


图 1 系统架构

Fig.1 System architecture

第 1 层为感知层,通过各种传感器、RFID 标签、GPS、识读器等感应器件采集智能设备的操作和监测数据。

第 2 层为接入层,将从感应器件获取的操作和监测数据,通过 REST 服务接口获取。针对智能设备产生的数据量大、并发性强的特点,采用负载均衡的策略对接入的数据进行处理。

第 3 层为存储层,应对不同的业务需求,存储层采用 MongoDB 和 MySQL 集群融合的方式,分别存储智能设备和平台管理数据。

第 4 层为服务层,由 REST 接口模块和数据挖掘模块组成。REST 接口模块采用 REST 架构向应用层提供数据查询的接口。数据挖掘模块按照特定的需求对设备数据进行分析处理。

第 5 层为应用层,包括 Web 访问应用和第三方平台的应用。Web 访问应用包括地址服务应用、设备基本信息应用、设备操作应用和统计分析应用。第三方平台可以根据需求和权限,获取相应的数据。

## 2 平台功能

从功能的角度来看,平台包括设备数据管理系统和商务服务系统。

### 2.1 设备数据管理系统

设备数据管理系统包括地址服务应用、设备基本信息应用、设备操作应用和统计分析应用 4 个方面的应用。地址服务应用是基于设备位置信息的一种应用。平台根据设备收集的经纬度信息,利用百度热力图展示各类设备在全国的分布情况,利用统计图表展示各类设备每天在全国的登录数量。设备基本信息应用是将每个设备在一天的所有操作进行汇总,以表格的方式展示了解设备每天的所有监测信息。设备操作应用是将人们对于设备的操作进行统计分析,以多种统计图的方式直观展示设备的操作情况。统计分析部分是对所有设备信息的汇总,如操作统计、检测值统计、时空统计等。

### 2.2 商务服务系统

商务服务系统提供对产品在线销售、APP 下载、技术服务、企业加盟、新闻动态、产品展示等商务服务的管理。通过该系统个人和企业都可了解产品信息并进行各种在线交易活动。

## 3 数据存储

物联网中传感器将实时产生海量数据,其存储分

析成为难点。MongoDB 作为一种基于文档的非关系性数据库,可以为物联网接入云服务平台提供可扩展、高性能的数据存储解决方案。

### 3.1 MongoDB 集群

本平台采用 sharding 集群技术搭建 MongoDB 数据库集群,针对海量传感数据的特点设置分片并按分片键均衡数据库的访问,而且可根据系统压力的增大,增加更多的分片提高系统的处理能力。

Sharde 分片集群包括 3 个组件:Shards、Config Servers、Query Routers。其中,Shards 用于存储数据,可以保证集群的高可用和数据一致性。Query Routers 即 mongos 实例,用于与应用程序交互,将请求转发到 Shards,然后将结果返回给客户端。一个集群可以有多个 mongos 实例用于分摊客户端的请求压力。Config Servers 用于存储分片集群的元数据,这些元数据包含了整个集群的数据集合与 Shards 的对应关系。Query Routers 使用这些元数据将客户端的请求转发到对应的 Shard 中。因为 Config Servers 中的数据很重要,所以将其放置到不同的服务器中,并且每个分片集群的 Config Server 不能共用,必须分开部署。

本平台在 3 台服务器上搭建 MongoDB 集群,包含 3 个分片主节点,每个主节点在另外的 2 个服务器中分别有一个副本和一个仲裁实现冗余,以保证在主节点出现故障的情况下将副本节点提升成主节点提供服务。集群还包含 3 个 Mongos 均衡客户端的请求,3 个 Config Server 保证客户端的查询在某个 config Server 失效的情况下仍能提供服务。本平台的 MongoDB 集群的框架如图 2 所示。

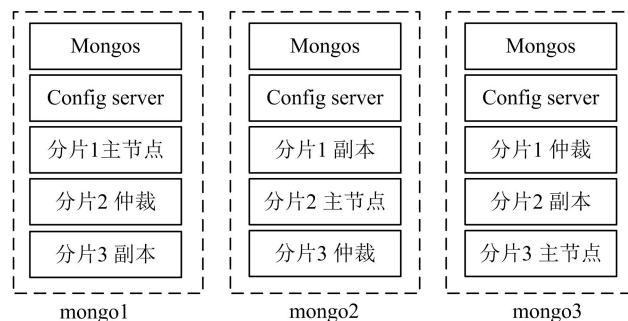


图 2 MongoDB 集群框架

Fig.2 MongoDB cluster framework

### 3.2 MongoDB 存储格式

平台根据海量物联网数据和智能设备的特点,为实现数据的高效存储和查询,将直接接入的数据存储在设备集合,而不同分析处理的结果存储在统计集合中。

物联网数据的异构性使得数据的表达非常困难,为实现数据的高效管理和共享,平台采用一种统一的数据表达方式来表达物联网数据。具体设计为:

Element= $\langle \text{key} ; \text{value} \rangle$ 。

其中:Element 为有序键值对; $\text{key} \in \text{Char}$ ,是 value 的名称; $\text{value} \in \text{Char} \cup \text{Number} \cup \text{Date} \cup \text{Element}$ ,用于存储实际的采样值,如 $\langle \text{longitude}; 116.331\ 398 \rangle$ , $\langle \text{latitude}; 39.897\ 445 \rangle$ 表示该设备现在所处的经度为 116.331 398,纬度为 39.897 445。记录由许多 Element 组成,主要包括 2 个部分:1)共有信息,如收集时间、收集经度、收集纬度等;2)特有信息,如智能灯的亮度、颜色、操作类型等。

### 3.3 MongoDB 访问

应用平台采用 Java 语言实现,利用 MongoDB 的 Java 驱动建立数据库连接。为更好地对数据库进行读写操作,保证数据操作的一致性,编写了一个 MongoDB\_driver 类。其主要代码为:

```
private static MongoClient mongoClient = null;
static{
    ..... //获取 MongoDB 连接属性
    private static MongoClient mongoClient = null;
    if(readStatus){
        try{
            ServerAddress serverAddr = new ServerAddress(mongo_ip, mongo_port);
            MongoCredential credential = MongoCredential.createCredential(mongo_username, mongo_dbname, mongo_password.toCharArray());
            List<MongoCredential> credential_list = new ArrayList<MongoCredential>();
            credential_list.add(credential);
            mongoClient = new MongoClient(serverAddr, credential_list);
        }catch (Exception e) {
            mongoClient = null;
        }
    }
    public static MongoClient getMongoClient(){
        return mongoClient;
    }
}
```

该类在静态代码块中获取配置文件中 MongoDB 的连接属性,通过布尔变量 readStatus 保证平台在同一时刻对数据库的连接只有一个,避免数据操作的不

一致性,通过静态函数 getMongoClient() 获得驱动类 MongoClient 实例,建立 MongoDB 数据库连接。

## 4 接入云服务

### 4.1 负载均衡

负载均衡(load balance)是将原独立的应用分配到多个操作单元上执行,解决系统的瓶颈问题。在大数据流式计算环境中,系统的负载均衡机制是系统稳定运行、高吞吐量计算、快速响应的一个关键。针对设备数据并发性能要求高的特点,本平台采用 DNS 负载均衡的策略保证系统的性能。通过在 DNS 服务器为多个地址配置同一个域名,使得对该域名的请求分配到不同的服务器中,达到负载均衡的目的。

### 4.2 REST 接入 API

REST 是由 Roy Fielding 博士提出的分布式系统的一种架构,它将网络上的任何具体信息,如一段文本、一张图片、一种服务等都看作是一种资源,这些资源通过统一资源标识符(uniform resource identifier,简称 URI)唯一标识,资源的操作通过 HTTP 协议中的 POST、GET、PUT、DELETE 等动作实现。

本平台采用 REST 架构实现数据接入层与服务层、服务层与应用层之间的交互,使得系统能灵活构建且各层的开发可以并行执行,提高开发效率。数据接入部分针对不同智能设备的特点,对每种类型的设备创建了一个类,该类的属性包括所有设备共有属性和该设备特有的属性。传感器将获取对应属性的监测数据,并通过 RESTful API 获得数据存入 MongoDB 的对应集合中。当有新的设备需要接入,只需要创建对应的类和其对应的 RESTful API,使得智能设备能灵活方便地接入系统。在数据应用部分,服务层向应用层提供获取设备数据的 API,包括获取数据记录的接口、不同分析结果的接口。接入层获取的数据和服务层提供的数据都是一种资源,可通过 URI 来表示。图 3 为本平台 REST 服务对应 URI 的表达式模板。

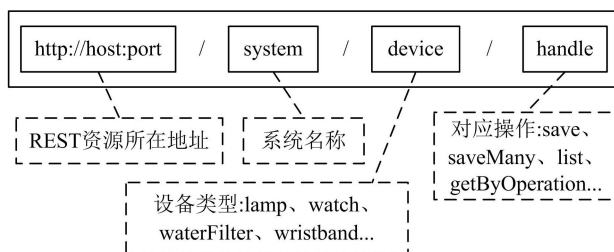


图 3 REST 服务对应 URI 的表达式模板

Fig.3 REST service corresponding to URI expression template

## 5 时空关系数据分析

本平台将物联网设备数据统计分析单独作为一个系统。系统采用批量处理的策略,将获得的数据采用相应的统计方法,获得需要的统计结果。以对智能设备的地址信息统计为例,在接入数据时,通过调用百度地图或者其他地图对应的地址解析接口,将获得的经纬度信息转换成所属的各级行政区。例如:(116.331 398,39.897 445)→(中国、北京、北京、东城区),并存入 MongoDB。在进行地址信息统计时,统计每天不同产品在各省、市、县的使用数量。具体算法为:

输入:该天收集到的所有数据  $X = \{x_1, x_2, x_3, \dots\}$ ;

输出:不同产品在各省、市、县的统计数据  $Y$ 、 $Z$ 、 $W$ 。

1)将  $X$  根据 MAC 去重,使得每个设备对应一条记录并计数为 1,结果如图 4(a)所示;

2)根据 1)的结果,将各产品属于同一个县的计数叠加,得到不同产品在各县的使用数量  $Y = n$  ( $n$  为该产品在该县的使用数量),结果如图 4(b)所示;

3)根据 2)的结果,将各产品属于同一个市的计数进行叠加,得到不同产品在各市的使用数量  $Z = \sum_{i=1, j=c}^n Y_{ij}$  ( $i$  为该市的各县,  $j$  为各市,  $n$  为该市中县的数量),结果如图 4(c)所示;

4)根据 3)的结果,将各产品属于同一个省的计数进行叠加,得到不同产品在各省的使用数量  $W = \sum_{i=1, j=c}^n Z_{ij}$  ( $i$  为该省的各市,  $j$  为各省,  $n$  为该省中市数量),结果如图 4(d)所示;

5)将 2)~4)的结果存入 MongoDB 的对应集合。

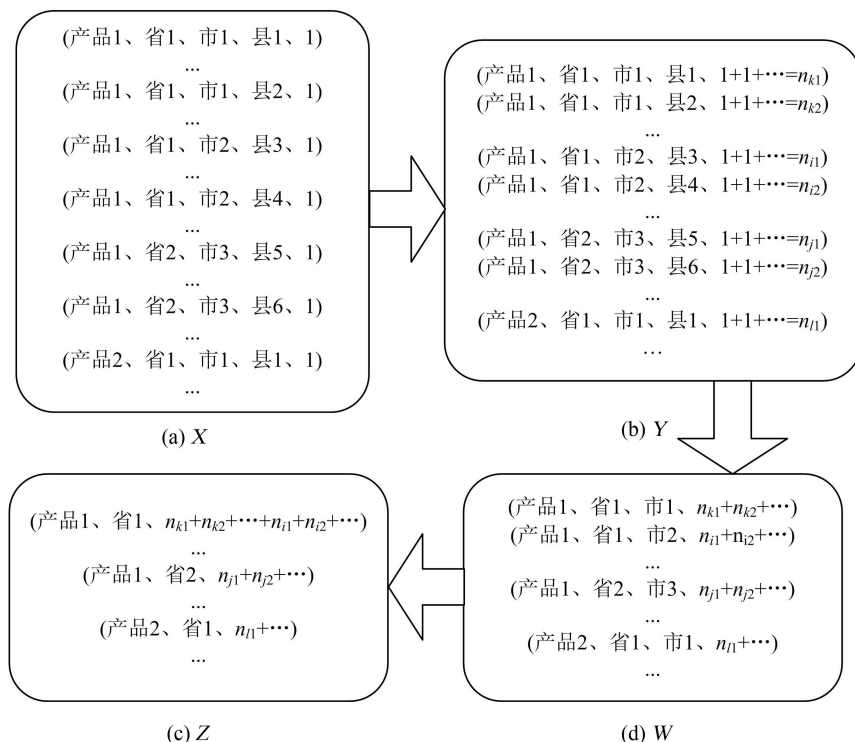


图 4 地址信息统计流程

Fig.4 Address information statistical process

同样地,在每月初的特定时间,按照省、市、县分别统计前一个月不同设备在对应行政区的使用数量,并将记录存储在 MongoDB 对应的集合中。使用地址信息统计,将海量的设备数据信息进行批量的统计处理,可以大大减少对数据库的访问压力,并能高效地获得查询结果。系统能根据统计结果提供产品分布、产品日统计、产品月统计等各种地址信息服务。

## 6 系统测试

### 6.1 实验环境

采用 Java 语言实现了整个基于 MongoDB 的智能物联云服务,在考虑成本和不失典型性的前提下,在 7 台安装 CentOS7 64 位操作系统的阿里云服务器 ECS 进行部署测试。其中,3 台部署 Restful 接入服

务器,3 台部署 MongoDB 数据库,1 台部署 Web 服务,除 Web 服务器为 2 核 CPU、2 GB 内存、1 MB 带宽外,其余 ECS 主机均为单核 CPU、1 GB 内存、1 MB 带宽。接入的设备为实际的智能灯、健康手环、摇摇车等实际试用设备。

## 6.2 功能测试

平台利用收集的智能设备数据的属性,采用相应的分析方法,统计分析对应的数据,获得数据的规律,挖掘数据的价值。图 5、6 为平台针对智能灯在数据分析上的展示。



图 5 智能灯全国分布

Fig.5 National distribution of smart lights

通过对收集到的传感器的时间和经纬度数据(时空数据),基于时空关系进行相应的处理分析,可得到每种设备每天在全国的各省的使用数量,再利用 echarts.js 的地图展示结果。图 5 为为智能灯某天在全国的使用情况。通过颜色的深浅表示使用数量的多少,颜色越深的省(区)表示使用该智能设备的数量越多。

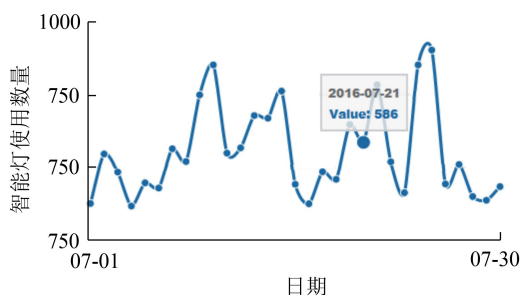


图 6 智能灯用户统计

Fig.6 Statistics on the number of smart lights users

设备使用情况的最直观表现就在于用户的使用数量。本平台通过对数据的分析统计,使用折线图展

示设备 2016 年 7 月的使用情况,并且用户可根据条件查询各个地区的使用情况。从图 6 生产商可直观地发现,用户在不同地区使用数量的总趋势,从而为商业决策提供参考。

## 6.3 性能测试

以智能灯的保存和查询设备数据接口为例,测试系统在基于 MongoDB 和 MySQL 存储时,各 RESTful API 的平均响应时间和响应时间标准差如表 1 所示。

表 1 RESTful API 测试结果  
Tab.1 RESTful API test results

接口	调用次数	平均响应时间/ms	响应时间标准差
lamp/save	15146	2.83	12.96
lamp/save_M	8880	106.56	715.73
lamp/get	1158	480.09	869.89
lamp_get_M	1116	485.09	853.793

表 1 中的 lamp/save 和 lamp/save\_M 接口用于存储智能设备的数据;lamp/get 和 lamp/get\_M 接口用于根据条件查询满足条件的结果。lamp/save、lamp/get 两个接口保存和查询的数据在 MongoDB, lamp/save\_M、lamp/get\_M 接口保存和查询的数据在 MySQL。从表 1 可看出,系统在基于 MongoDB 存储时,存储智能设备数据所需时间约为基于 MySQL 存储的 1/37.65,同时也能保证查询的效率,其标准差的对比也表明,基于 MongoDB 存储系统的稳定性更强。

## 7 结束语

为采集和分析海量智能设备数据,设计了一种基于 MongoDB 的物联网接入云服务平台。该平台利用 REST 服务接口,将传感器的采集数据接入到云服务平台,存入到 MongoDB,使用智能分析策略分析数据,通过 REST 数据服务接口的方式为应用层提供服务,能够将结果以可视化的方式展示。实验结果证明了平台在高并发量下的有效性。系统的长期应用实践也表明,本平台能够实时反映设备的使用和运行情况,分析挖掘设备信息的有用价值,为厂商的商业决策提供支撑。下一步将采用更多的智能算法和大数据处理策略为用户挖掘更多有价值的信息。

## 参考文献:

- [1] CONTI P. The Internet of Things[J]. Communications Engineer, 2006, 4(6): 20-25.
- [2] ATZORI L, IERA A, MORABITO G. The Internet of Things: a survey [J]. Computer Networks, 2010, 54(15): 2787-2805.
- [3] KORTUEM G, KAWSAR F, FITTON D, et al. Smart objects as building blocks for the Internet of Things[J]. IEEE Internet Computing, 2010, 14(1): 44-51.
- [4] 韩海雯, 齐德昱, 封斌. 基于云计算与物联网技术的港口物流综合服务平台架构研究[J]. 计算机科学, 2013, 40(s1): 232-235.
- [5] ZHANG Guigang, LI Chao, ZHANG Yong, et al. Semantic Medical: a kind of semantic medical monitoring system model based on the IoT sensors[M]//Proceedings of the IEEE 14th International Conference on e-Health Networking, Applications and Services. USA: IEEE, 2012: 238-243.
- [6] MAINETTI L, MIGHALI V, PATRONO L. A software architecture enabling the Web of things[J]. IEEE Internet of Things Journal, 2015, 2(6): 1-1.
- [7] OH S W, KIM H S. Study on access permission control for the Web of things[C]//International Conference on Advanced Communication Technology, 2015: 574-580.
- [8] GONZALEZ H, HAN Jiawei, LI Xiaolei, et al. Warehousing and analyzing massive RFID data sets[M]//Proceeding of the 22nd International Conference on Data Engineering, 2006: 83.
- [9] YANG Hui, QIN Yong, FENG Gefei, et al. Online monitoring of geological CO<sub>2</sub> storage and leakage based on wireless sensor networks [J]. IEEE Sensors Journal, 2013, 13(2): 556-562.
- [10] 丁治明, 高需. 面向物联网海量传感器采样数据管理的数据库集群系统框架[J]. 计算机学报, 2012, 35(6): 2514-2517.
- [11] LI Tingli, LIU Yang, TIAN Ye, et al. A storage solution for massive IoT data based on NoSQL[M]//Proceedings of the IEEE International Conference on Internet of Things, 2012: 50-57.
- [12] YIN S, CHEN D, LE J. STNoSQL: Creating NoSQL database on the SensibleThings platform [C]//IEEE Ieee/acis International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2016: 669-674.
- [13] FRANCESCO M D, LI Na, RAJ M, et al. A storage infrastructure for heterogeneous and multimedia data in the Internet of Things[M]//Proceedings of the 2012 IEEE International Conference on Internet of Things, 2012: 26-33.
- [14] 田野, 袁博, 李廷力. 物联网海量异构数据存储与共享策略研究[J]. 电子学报, 2016, 44(2): 247-257.

编辑: 梁王欢