

物联网海量异构数据存储与共享策略研究

田 野¹,袁 博¹,李廷力²

(1. 中国科学院计算机网络信息中心物联网信息中心,北京 100190;2. 北京百度网讯科技有限公司,北京 100085)

摘 要: 随着物联网向各行业的深入发展,各行业的信息化进程也进入了快车道. 信息服务作为物联网在各行业应用中重要的公共服务之一,一直受到广泛关注. 然而,当前物联网信息服务系统面对物联网海量异构数据存在性能低下、共享困难等问题. 因此,本文提出了一种基于 NoSQL、REST 以及国家物联网标识管理公共服务平台(NIOT)的存储与共享策略,并着重对该系统的构成、逻辑设计进行了详尽阐述. 针对性能改进的策略设计了适当的量化评测,实验结果表明提出策略具有较好的效果,基于实验结果对进一步的优化进行了讨论.

关键词: 物联网; 海量异构数据; 信息服务系统; 数据存储; 数据共享

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2016)02-0247-11

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.02.002

A Massive and Heterogeneous Data Storage and Sharing Strategy for Internet of Things

TIAN Ye¹, YUAN Bo¹, LI Ting-li²

(1. Computer Network Information Center, Chinese Academy of Science, Beijing 100190, China;

2. Beijing Baidu Netcom Science and Technology Company, Ltd, Beijing 100085, China)

Abstract: With the development of the Internet of Things (IoT), it accelerates the process of the informatization of all industries. Information service is an important service in the IoT. However, massive and heterogeneous data of the IoT brings the storage huge challenges to the information service. This paper proposes an IoT information service system to solve the current problem of the performance shortfalls and the difficulty of data sharing, which is based on NoSQL, REST and NIOT (National IOT id management and public service platform). The work focuses on improving the performance of the system and exploring the expression of storage and the sharing mechanism of the heterogeneous data. The test result shows that the solution proposed by this paper significantly enhanced the system performance. The relevant achievements of this paper provide reference for facilitating the development of the IoT information service system.

Key words: internet of things; massive and heterogeneous data; information service system; data storage; data sharing

1 引言

物联网虽然已在物流、交通、电力等行业形成了一定规模的应用,但整体还处于发展初期,其信息化、智能化程度仍然较低,需要专门的研究以突破物联网信息化、智能化进程中的难题.

从信息处理视角出发,如图1左边所示,物联网至下而上可分为四个层次:感知层、网络层、数据层以及应用层. 信息服务系统位于数据层,它向下接收海量多源异构的数据,向上服务于各类规模应用,提供多样化的

信息服务,起到承上启下的作用,是实现物联网互联互通的前提;基于信息服务系统,可使物联网应用具备对现实世界各类物体的信息采样、智能追踪、智能监控和智能管理等功能,因此,物联网信息服务系统是物联网信息化进程中不可或缺的基石.

作为物联网信息化、智能化的重要手段,物联网信息服务系统是一个重要的研究方向. 其结构如图1右边所示,物联网信息服务系统主要由持久化存储系统、数据接口、服务操作、安全机制、共享机制等部分构成;在

收稿日期:2014-04-23;修回日期:2015-08-26;责任编辑:蓝红杰

基金项目:国家科技支撑计划(No. 2015BAK36B02);国家发改委2012物联网技术研发产业化专项(物联网标识管理公共服务平台);中国科学院一三五规划重点培育方向(No. CNIC_PY_1403)

功能上,该系统目标是存储并管理物联网数据,支撑不同规模数据(实时的、历史性的)的信息处理,并提供良好的开放性,支持不同物联网应用之间实现数据共享。

基于该系统,物联网应用可实现对现实世界各类物体的信息采样以及智能追踪、监控和管理等功能。

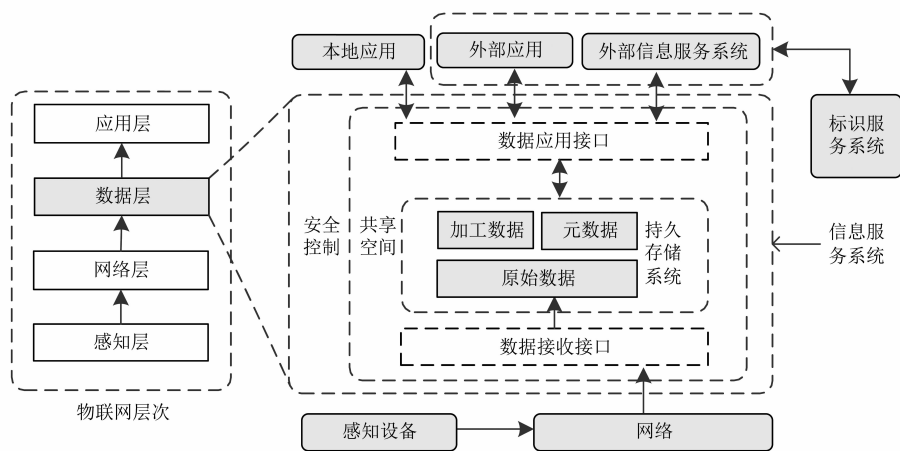


图1 物联网层次与信息服务系统

物联网数据具有多源异构、规模巨大、时空关联、冗余度高、多维标量等特性^[1],现有物联网信息服务系统解决方案在时空关联处理,多维事件查询等方面取得了较多成果,但在面对海量异构数据时还存在存储性能差、数据共享困难的问题。数据存储问题主要体现在老式关系数据库存储模式不能满足多源、异构、海量的物联网数据存储需求。数据共享问题主要体现在缺乏有效共享机制解决物联网应用闭环,容易形成信息孤岛。针对以上两个问题,本文开展物联网信息服务系统海量异构数据的存储与共享策略研究,提出一种基于非关系型数据库(Not Only SQL, NoSQL)技术的存储方案和策略,实验证明,该方案提高了海量数据下物联网信息服务系统的性能;同时探索了异构物联网数据的共享机制,助力于打破物联网普遍存在的应用闭环。

2 相关工作

2.1 物联网海量数据存储

数据存储是物联网信息服务系统的核心,许多工作都基于不同技术(文件系统、数据库)开展对物联网海量数据存储的研究,并取得不少成果。

(1) 基于文件系统

多数研究关注使用分布式文件系统存储物联网海量数据^[2,3],这类文件系统建立于本地文件系统上,通过网络将若干存储节点相连,逻辑上将独立的存储节点聚合为一个文件系统整体,统一管理节点物理资源,同时提供并发控制实现多用户文件访问,可以解决 I/O 瓶颈、空间动态扩展等问题。

然而,无论是本地文件系统还是分布式文件系统,都存在如下问题:文件结构与应用紧耦合导致数据共

享困难;数据冗余度大导致数据不一致;大量物联网小数据^[2]需要专门处理方法^[4,5];物联网数据的多维度、多粒度特性导致数据检索困难;文件系统处理实时在线数据流处理能力较差。

因此,基于文件系统的方案并不适用于存储物联网海量数据。

(2) 基于数据库技术

数据库技术从数据模型角度主要分为两大类:关系型数据库和非关系型数据库。

关系型数据库技术成熟,多数物联网海量存储方案都选择这种技术为基础^[6,7]。然而,简单将数据库推广至物联网很难满足其需求,因此一些研究针对物联网数据的特性提出了相应地解决方案,文献[8]在分析物品移动路径特征与采样数据实际应用状况的基础上,针对物流仓储提出了一个面向 RFID 海量数据的新数据模型 RFID-Cuboids,既完整保存了物品状态变化,又对数据进行了显著压缩。由于上述存储方案仅针对物联网某种技术或业务领域,应用场景局限性大。因此,文献[9]面向整个物联网,特别针对物联网数据时空关联特性,在改良 RDBMS 基础上提出了兼容键值(Key-Value)查询方式的数据库模型 RDB-KV。该方案对于重复度较高的数据,在保证数据正确性的前提下减少了数据量,提高了数据质量。然而该方案核心技术为 RDBMS,它处理物联网数据存在海量数据存储困难、不支持模糊查询、不满足数据流实时查询需求等不足。

因此,一些新兴的数据库技术得到了不少关注,尤其是 NoSQL 技术。该技术由于性能、扩展性方面的优势已在互联网中处理大数据方面有较成功的应用。较为流行的开源 NoSQL 数据库有 MongoDB、Redis 以及 Cas-

sandra 等;应用较广的非开源数据库有谷歌的 BigTable 以及亚马逊的 Dynamo 等。

已有一些工作将 NoSQL 数据库应用于物联网领域,如文献[10]将 NoSQL 应用构建物联网异构多媒体数据存储架构.但由于 NoSQL 技术种类繁多,该方案并未给出合理的选择依据.此外,已有方案对物联网感知数据的时空关联性以及数据不确定性方面并未进行研究,对于如何使用 NoSQL 应对物联网数据的种类特性研究深度不够。

(3) 混合技术

部分应用出于自身考虑设计了数据库与文件系统结合使用的方式。

文献[11]针对智能电网数据存储,设计了 RDBMS 与本地文件系统结合的存储架构,将数据存储于文件系统中,而文件路径及时间信息则存储于数据库.该方案本质上是对本地文件系统方案的改进,提高文件检索效率.但这种改进效果不大,其仿真实验结果表明,其执行时间长、性能差.文献[3]使用分布式文件系统 HDFS 与数据库结合构建智慧医疗应用,对不同格式数据进行了分类存储,借助数据库的强大查询功能支持语义规则分析。

混合式可以利用数据库与文件系统各自的优点,但由于涉及两类存储系统,在访问数据时比单一系统增加了额外的开销,系统性能也会受到影响。

2.2 物联网数据共享

在物联网数据共享方面,数据交换标准定义了物联网数据交换协议与交互方式. PML(Physical Markup Language,物理标识语言)、EDDL(Electronic Device Description Language,电子设备描述语言)、M2MXML 及 NGTP(Next Generation Telematics Protocol,下一代远距离通信协议)是物联网中主流的应用层数据交换标准,应用于某特定行业、领域或业务.由于各标准本身的局限性,难以通过它们实现不同物联网应用之间的无缝连接与互联互通,这也正是物联网闭环产生的重要原因之一。

因此,有大量研究解决感知网与互联网数据共享问题.框架方面,文献[12]基于 DNS 提出物联网数据共享框架,但仅是理论探讨;技术方面,受限应用协议(Constrained Application Protocol, CoAP)^[13]为受限网络和节点制定符合 REST 风格的应用协议,通过特定网关实现 CoAP 协议与 HTTP 协议的映射,达到物联网与互联网进行数据共享目的.此外,文献[14]研究了如何使用 REST 接口共享传感数据,文献[15]研究了 REST 在 RFID 网络中的应用.因此,REST 更有利于实现物联网异构数据共享。

总体而言,当前的物联网信息服务系统的研究存在两个突出的问题:面向物联网海量数据,存储性能较

低;面向物联网异构数据,缺乏有效的共享机制,导致了物联网应用闭环的广泛存在。

3 面向物联网海量异构数据的存储方案及策略

3.1 存储技术选择

3.1.1 定性分析

主流存储技术有 RDBMS 和 NoSQL, LNoSQL 在性能、扩展性和异构数据处理更优,对数据一致性要求不高,更加符合物联网需求.从数据模型的角度, NoSQL 可分为四类:(1)键值(Key-Value)型,通过某种方式为值建立索引,值通过一个唯一的键进行检索,典型代表为 Redis、Membase、Voldemort;(2)文档型(Document),将 key-value 对封装到 json 或类 json 文档中,文档可嵌套,典型代表有 MongoDB、CouchDB、Riak;(3)列式(Column Family),存储数据以“列”为单位,典型代表有 Cassandra、Hbase、Hypertable;(4)图式(Graph),管理有复杂密集关联的数据,适于基于关系复杂数据的应用,典型代表有 Neo4J、GraphDB 等。

下面对 NoSQL 的 Redis、MongoDB、Cassandra 以及 RDBMS 的 MySQL 进行评测分析:MySQL 适合在线事务处理应用,但由于其扩展性差,不适合物联网海量异构数据存储;Redis 是纯内存操作,性能高,但海量数据很难全内存处理;MongoDB 不仅支持海量数据存储,还有有效解决海量数据访问效率问题,且支持复杂数据结构,能有效应对异构数据;Cassandra 是由一堆数据库节点共同构成的一个分布式网络服务,也适用于海量数据存储,其数据接口与 MongoDB 类似,但查询稍弱.因此, MongoDB 与 Cassandra 能满足物联网数据存储需求,能支持海量数据背景下物联网信息服务系统的构建。

3.1.2 定量分析

本文设计了如下性能测评,为确定具体数据库技术提供量化依据.评测包括:

(1)并发写测试,分别使用 1、2、4、8、16 个线程向数据库插入数据,每个线程共插入 10 万条数据,每次写入一条.并发写性能计算方法如下:

$$P_{\text{insert}} = \frac{N_{\text{thread}} * N_{\text{insert}}}{t_{\text{max}}} \quad (1)$$

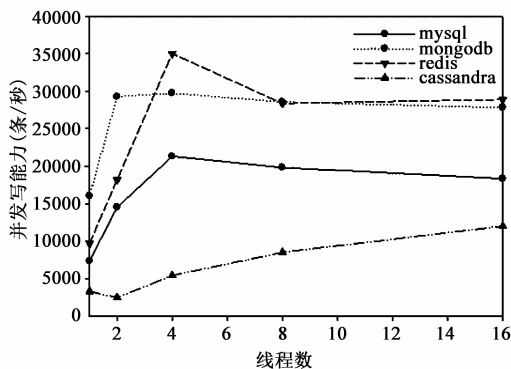
其中, P_{insert} 表示插入性能,计量单位为“条/秒”, N_{thread} 表示线程数, N_{insert} 表示每条线程插入记录数, t_{max} 表示最长耗时.所有测试用例执行 5 次,结果为去掉最大、最小值后取平均值。

(2)并发读测试,分别使用 1、2、4、8、16、32、64 个线程对各个数据库执行查询.查询内容等效于如下 SQL 语句:

```
select * from dataset where moteid = 1 and date = '
```

2004-03-22';

读测试数据规模约 5000 万条,结果集约 5 万条.由于查询规模大,一次查询耗时长,因此并发读性能以查询时延衡量,结果取各线程最长耗时,测试用例同样执行五遍并按前文所述方法取平均值.图 2 展示并发读



写的性能评测结果,其中 Redis 性能最优,MongoDB 其次.

综上所述,MongoDB 最适用于支持物联网信息服务系统建设,虽然性能较 Redis 低,但在应对海量数据方面具备更大优势.

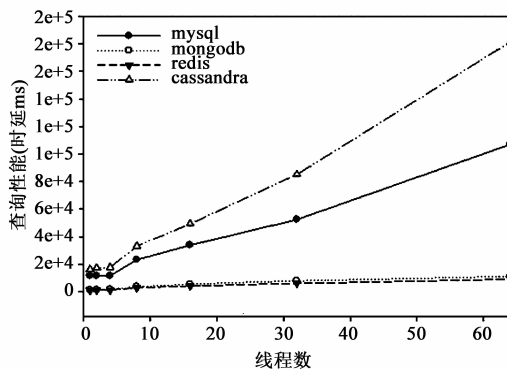


图2 数据库系统性能评测结果

3.2 存储策略

本文针对海量异构物联网数据提出了存储策略,包含:预处理、统一数据表达方式、数据分布等.在预处理中,将物联网数据分为轻量级数据和多媒体数据,并将多媒体数据进行轻量化处理存储,减少存储数据量,提高数据质量;统一数据表达方式使得物联网数据存储更规范,易于共享;数据分布主要从数据和数据库层面入手,改善存储效率.

3.2.1 知识密度

下面先定义“知识密度”概念.

定义 1 知识密度. 指尽可能完整表述某特定信息所需的数据量与原始数据量之比,用小数表示.

例如,一段音频最高噪声为 85 分贝,这个“知识”用字符串表示: < max_noise:85 >

假设音频大小为 1KB,而该字符串 14 个英文字符,占 14 字节,则其知识密度为 0.0137(14/1024).

知识密度大小由具体关心的知识决定.仍以音频为例,若要获得不同时间段噪声大小,通过处理手段得出噪声序列 {16, 23, 85, 45, 61}, 仍用字符串表示: < noise_seq:16,23,85,45,61 >

该知识密度为 0.0254(26/1024),比之前高.

3.2.2 预处理

数据预处理由数据接收节点完成,处理过程如图 3 所示,主要有四个处理阶段,其中如下两个阶段是本文提出的预处理策略中的核心:

(1) 数据分类

原始采样数据被分为两类:轻量级数据,包括数值型及字符型,其特征是占用空间小,传输开销少;多媒体数据,包括视频、图像、音频以及信号等,其特征是存

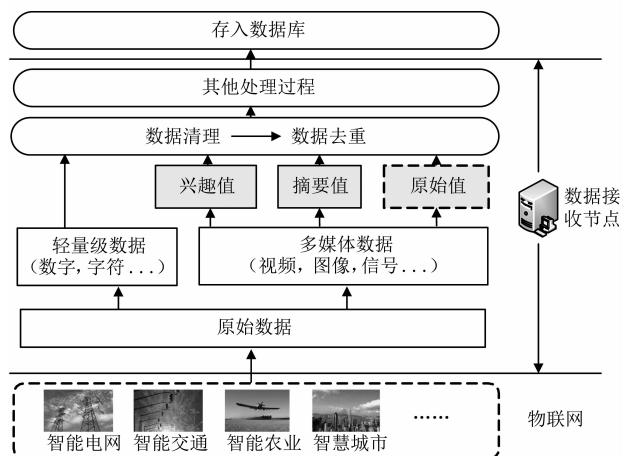


图3 预处理过程

储空间和计算资源需求大.在预处理中,针对这两类不同数据处理方式各异.

(2) 增稠知识密度

增稠知识密度是存储策略中的重要处理步骤,主要针对多媒体数据.多媒体数据占用存储空间大,知识密度低,需要轻量化处理,提取有价值部分,将关键知识以较低数据量进行存储,提高数据应用效率,其原始数据可根据实际需求确定是否保存.

根据实际应用需要,轻量化多媒体数据时需要提取出两类数据值,最大化保存数据蕴含信息:兴趣值,通过算法提取应用最关心的信息,如智能交通系统需要从监控视频中分析出一小时内交通违规数量及违规车辆的车牌号等;摘要值,同样由特定算法计算(如 MD5),用于对多媒体数据进行简要描述,其知识密度几乎为 0,是固定长度字符串,用于辅助多媒体数据去

重与检索,如判定两幅图是否一样可通过比对摘要值。

3.2.3 数据表达

物联网数据的异构性为数据表达带来了困难,标准化表达方式有助于降低数据存储难度,提高数据管理效率,方便数据共享.本文设计一种统一方式实现异构物联网数据的表达,整体设计如下:

结合 NoSQL 数据库数据格式特点,信息服务系统中原子存储单元定义为 SampleElement:

$$\text{SampleElement} = \langle \text{key}; \text{value} \rangle \quad (2)$$

SampleElement 是有序键值对,其中 $\text{key} \in \text{Char}$, 是 value 的名称;而 $\text{value} \in \text{Char} \cup \text{Number}$, 用于存储实际

采样值,实例如下: $\langle \text{temperature}; 50 \rangle$ 或 $\langle \text{audioText}; \text{"Hello world"} \rangle$, 表示采样温度 50℃, 从音频中提取文本“Hello world”。

如图 4 所示,信息服务系统中基本存储单元是一条记录,即 SampleRecord, 由 SampleElement 构成的集合. SampleRecord 包括两部分信息:静态信息,如对象 ID、所属领域、物体类型等,是物体的特征信息,基本保持不变,只需存储一次;动态信息,即物体实际采样值,如时间、地点、状态及环境信息,反映物体变化情况,由轻量级数据和稠化知识密度的多媒体数据两部分构成。

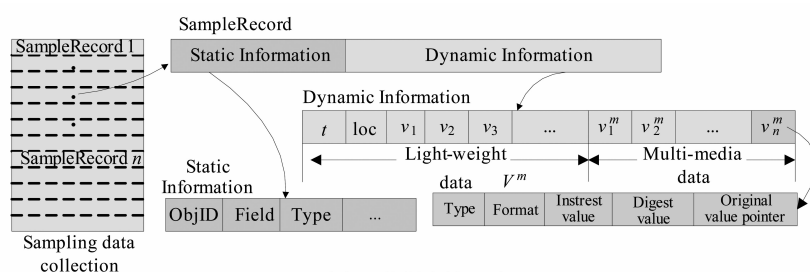


图4 数据表达形式

SampleRecord 具体定义如下:

$$\text{SampleRecord} = (\text{rID}, \text{sID}, \text{tID}, \text{field}, \text{type}, \dots), \quad (3)$$

$$(t, \text{loc}(v_i)_{i=1}^n, (v_j^m)_{j=1}^x)$$

其中, $\text{rID}, \text{sID}, \text{tID} \in \text{String}$, rID 是记录的 ID, 具有唯一性, tID 是物体的物品码, sID 是物品码所属的标准码. 在物联网领域内, 标识编码存在多种体系, 如 EPC 编码体系、OID 编码体系等, 因此需要明确标识所属标准进行区分. $\text{field} \in \text{String}$ 用于标识物品所属领域; type 等是物体的其他静态描述信息; $t \in \text{Instant}$ 是采样时间; $\text{loc} \in \text{Point}$ 是采样地点, Point 是二维值如 (x, y) , 可自定义空间; $v \in \text{String} \cup \text{Number}$ 是采样的轻量级数据; $v^m \in \text{SampleMedia}$ 用于表达预处理后的多媒体数据, 定义如下:

$$\text{SampleMedia} = (\text{type}, \text{format}, v_{\text{interest}}, v_{\text{digest}}, p_{\text{original}}) \quad (4)$$

其中, $\text{type}, \text{format} \in \text{String}$, 分别表示该多媒体数据类型 (如音频、图像或视频) 及格式 (如 jpeg, gif, mp4 等); $v_{\text{interest}} \in \text{String} \cup \text{Number}$ 指从多媒体数据中提取的兴趣值; $v_{\text{digest}} \in \text{String}$ 指轻量化多媒体数据后提取的摘要值; p_{original} 是指向多媒体数据原始值的指针。

如图 4 所示, 一个集合由若干条 SampleRecord 组成, 一条 SampleRecord 可视作 RDMBS 表中的一行. 通过 SampleRecord 可实现对物联网异构数据的统一表达, 并有助于数据的共享。

下面实例化介绍 SampleRecord 各部分内容. 示例 1: $(\langle \text{objID}; \text{"a01"} \rangle, \langle \text{field}; \text{"agriculture"} \rangle, \langle \text{type}; \text{"shed"} \rangle), (t; t1, \text{loc}; (40.1, 20.5)), (\text{light}; 50, \text{humidity}; 0.56, \text{light}; 23.1))$, 表示对物联网农业领域应用中大棚的采样数

据, 采样时间为 t_1 , 地点为 GPS 坐标 $(40.1, 20.5)$, 光照度为 50 勒克斯, 相对湿度为 56%, 湿度为 23.1 摄氏度. 示例 2: $(\langle \text{objID}; \text{"b123"} \rangle, \langle \text{field}; \text{"traffic"} \rangle, \langle \text{type}; \text{"car"} \rangle), (t; t2, \text{loc}; (123, 1024), \text{speed}; 210, \text{capturePhoto}; (\text{type}; \text{"image"}, \text{format}; \text{"jpeg"}, \text{plateNumber}; \text{"BJ0A435"}, \text{digestValue}; \text{"b38767a34dd2d764c0d597986010e5b1"}, \text{originalValue}; \text{"p1"}))$, 来源于交通领域的超速检测, 采样地点为 $(123, 1024)$, 这是自定义空间坐标, 超速车辆速度是 210km/h, 记录中包含一个多媒体数据 capturePhoto, 指抓拍的超速车辆照片, 格式为 jpeg, 从照片中提取的兴趣值为车牌号 BJ0A435, 摘要值“b38767a34dd2d764c0d597986010e5b1”是 MD5 值, 指向该照片的原始数据指针为 p_1 。

3.2.4 数据分布

数据分布分为数据层面和数据库层面. 数据层面, 考虑数据的逻辑分布; 数据库层面, 从数据物理分布入手, 研究数据是否应该分布到不同实体机器/或不同的片存储, 以及数据分布的策略。

(1) 数据类型

对于数据类型, 将多媒体数据与轻量级数据分离存储. 为提高集群性能, 多媒体数据原始值与轻量化数据分离存储. 若需要获得多媒体数据的原始值, 可使用 SampleMedia 中的原始数据指针。

分离存储有两种选择, 包括: 设置主副集群, 两个集群各运行一个数据库实例, 将多媒体数据原始值存储于副集群中, 外部用户对副集群无感知, 两个集群之

间通过某种方式通信获得数据;将多媒体数据与轻量化数据存储于不同集合中,但逻辑上仍在同一个数据库中.第一种选择拥有独立的计算能力应对多媒体数据的需求,但通信会带来额外开销;第二种选择不会造成额外通信开销,但由于分享共同的计算资源,在性能上会造成一定的影响.

(2) 数据库

由于系统采用分布式架构,当数据量增加到单个机器无法完全存储时,必须进行数据分片^[16],分散到不同机器存储,使得集群可不依靠强大服务器实现海量数据存储.

分片有助于提升存储能力和服务器性能,但并非一定能提升性能,如果应用不合理反而会造成性能损耗.在分片的情况下,一个查询的执行时间 T_q 如下公式所示:

$$T_q + 2 * T_t + T_p + \frac{T_s}{N_p} + T_a \quad (5)$$

其中, T_t 是客户端与服务器的传输时延; T_p 是解析查询指令并将指令分发给从节点的时间; T_s 是查询指令在单台机器的执行时间; T_a 是从多个节点中汇集查询结果的时间,与并行执行指令的机器数据 N_p 成正比.通常 T_t 与 T_p 要远小于 T_s 与 T_a ,因此 $T_q \approx \frac{T_s}{N_p} + T_a$.如果并行执行指令节省时间少于汇集结果引发的额外开销,则整体执行时间将增加.

因此,对于规模较小的数据不必分片,以避免存储空间扩展带来的时间开销.在数据量增大的情况下,分片带来的附加性能开销是小于数据量较小时分片带来的开销的.

4 共享机制

数据共享主要解决在信息服务系统中数据抽象、数据定位和数据获取三个问题,本节从这三个问题入手,介绍在物联网数据共享方面的研究,主要分为两部分:基于 REST 的信息服务解决数据抽象和获取,基于国家物联网标识管理公共服务平台 (NIOT)^[17-19] 解决数据定位.

4.1 基于 REST 的信息共享服务

REST 是一种设计准则,目的是指导网络程序开发,降低开发复杂性,提高系统可伸缩性. ROA 是 RESTful 架构中的典型实例,它将实际问题转换成 REST 描述,形成一种明确、简单易用的架构.本文采用 ROA 架构设计并实现物联网信息服务系统.

4.1.1 资源定义

在 REST 设计原则中,一切可以被命名的实体都可被抽象为资源,资源不但包括传统网络中的信息、链

接、计算、存储等,更包括物联网中接入网关、感知网络、感知数据等一系列信息.

本文设计的物联网信息服务系统将资源分为三类:对象资源、索引资源和知识资源.其中,对象资源指单个物体对象对应的资源,是单条数据记录或数据记录的集合;索引资源指某一类或某个集合资源的索引,通过该资源可以获得资源列表,进而实现对对象资源的访问;知识资源指通过某种手段或算法(如统计、过滤、图像识别等)基于对象资源进行操作所生成的资源,通常代表某种知识.

为了方便资源管理,本文对资源进行层次化管理方式,分为实体级、应用级和系统级.其中,实体级指在此级别上访问的资源都是单个实体资源,对象资源、索引资源、知识资源都属于实体级资源;应用级指若干实体资源的集合,一个应用中存在多个感知对象,属于该应用的所有对象资源都被划归为同一个集合中;系统级指信息服务系统层级的资源,信息服务系统层级由若干个应用组成.

图 5 展示了资源层级与资源类型间的关系.资源间的关系通过索引资源的标识发生状态转移获得对象资源及知识资源.对象资源标识可通过状态获得感知数据或知识资源.知识资源通过一定方式可进行知识提取,获得生成数据.图 5 概要展示了资源类型间的关系,实际上,各类资源还可以有更多层次的转移路径,如某索引资源可以转移到另一索引资源,进而实现对不同类型资源的灵活组织.

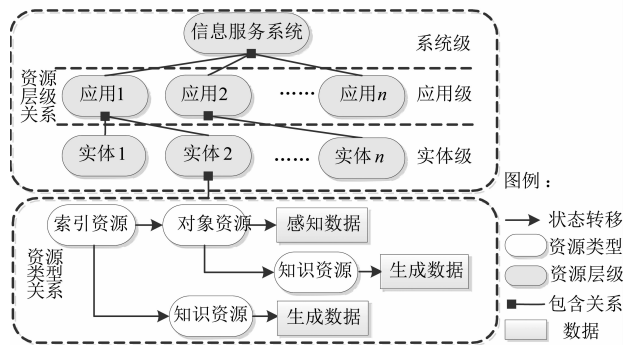


图5 资源层级与资源类型关系

4.1.2 资源标识

使资源具备开放性的方式是构造资源标识并对外发布,使得资源可以被寻址定位,本小节将详细阐述资源标识的构成、设计.

(1) 标识概念

基于 ROA 的信息服务将为每个资源分配一个全局唯一的资源标识符 URI,并对资源提供全局共享请求消息分类系统.信息服务系统接收到应用程序的资源请求后,先将请求分解,并将分析结果进行语义处理,然

后得到一个 URI 列表,并对被请求 URI 进行解析处理,最终将该资源决策及相关资源通过超文本形式资源表示发送至应用程序.由于 REST 架构是无状态的,因此每个请求都须包含理解该请求所需的全部信息.如一个温度资源查询请求,URI 格式如下,标识了中国科学院计算机网络信息中心智能家庭应用中 1 号楼 702 房间的温度资源:

```
http://www.niot.cn/smarthome/building4/room402/temperature
```

(2) 标识构成

资源描述是信息服务系统进行资源管理的基础,URI 描述的资源应表现资源相关特性.本文设计的 URI 描述采用与资源本身信息和属性结合的方式,即不同 URI 的同一级别表示同一类属性.重点考虑如下属性:网络名,资源所在网络的名称(如 sina、unicom 等);应用领域,使用资源的应用所在领域(如交通、农业等);资源类型,资源属性的分类(如温度、湿度等);资源 ID,资源名称或编号;位置信息,标明资源所在位置,可多级表示.

(3) 标识设计

本文设计 URI 层次资源管理方式,URI 的根是系统级资源,通过应用 ID 获得应用级资源;在应用级资源中,通过实例资源 ID 获得实例级资源(对象资源、索引资源、知识资源).不同层级之间通过索引资源实现访问,在索引资源中可以通过单个对象 ID 向下级访问获得对象资源,索引资源中提供有限的只读操作,如果需要对象资源进行编辑,需要通过鉴权等方式获得资源的编辑权限.

表 1 给出资源标识具体设计,其中“/”代表系统级资源;“Application”代表应用级资源,通过“id”获得实例级资源,实例级资源可以相互嵌套;“oper”指预设的操作(如统计等)用于生成知识资源;“data”指感知数据或生成数据.

表 1 资源标识设计

Method	Application	Resource	Data
POST	/application	/application/ {id}/resources	/application/ { id }/resources/ {data}
GET		/applications/ { id }/resources	/applications/ { id }/resources/
PUT	/applications/ {id}	or/applications/ { id }/resources/	{ id } or/applica- tions/{ id }/re-
DELETE		{oper}	sources/{ id }/o- per}

表 1 中规定了对资源进行的操作,POST 创建资源,GET、PUT、DELETE 表示获取、更新、删除资源,其他操作通过自定义“oper”实现.

4.1.3 资源表述

ROA 架构下,URI 具备多重表述的特性,可根据实际应用需求生成不同格式的数据,都是“逻辑”URI.逻辑 URI 与传统的 WEB 应用中的“物理”URI 区别在于逻辑 URI 没有格式后缀.当资源表述形式改变后,物理 URI 也要变更,已发布 URI 将失效.而逻辑 URI 则无需变更,可永久保持其有效性.

逻辑 URI: http://www.datacenter.com/data/10

物理 URI: http://www.datacenter.com/data/10.html

不同资源表述(如 JSON、XML 等)都可传输数据,但选择何种资源表述需综合考虑性能、带宽占用以及可读性等因素.本文选择 XML 和 JSON 进行评测,测试不同资源表述对信息服务系统额外开销,为资源表述方式选择提供量化参考.

测试内容为:在不同字段数、不同有效载荷下,传输时延和存储开销.结果如表 2 所示,在相同有效载荷下,XML 比 JSON 存储开销大,且处理与传输时延大于 JSON.

表 2 不同数据交换格式性能对比

字段数	有效载荷	XML 数据	JSON 数据
3	137 Byte	278 Byte,4.440ms	206 Byte,4.414ms
3	1049 Byte	1494 Byte,4.922ms	1422 Byte,4.548ms
4	1056 Byte	1516 Byte,5.077ms	1440 Byte,4.571ms
8	1157 Byte	1600 Byte,5.622ms	1505 Byte,4.748ms

在物联网应用中,数据是多维或高维的,JSON 比 XML 更具空间优势.但是,JSON 可读性差,更适用于 M2M 应用.

4.1.4 标准接口

资源被发布、定位后,需要与外界交互.标准接口用以减少交互成本,方便数据共享.

基于 REST 风格的信息接口,交互都可由若干 HTTP 标准请求实现,本文定义了 GET、PUT、DELETE 以及 POST 以完成对资源的 CRUD(Create、Retrieve、Update、Delete)操作(详见表 1).

另一方面,REST 对系统进行了层次化划分,增加了层间数据处理开销和延迟,同时 HTTP 协议据包头也会降低系统性能.本文对 REST 应用开展性能评测(如图 6 所示),评测使用 RESTful 数据接口与原生数据接口的信息服务系统,在不同数据大小时测试数据传输时延.结果表明采用 REST 架构增加了数据传输延迟,但仍在可接受范围内.

4.2 基于 NIOT 的信息共享服务

物联网拥有海量资源,如何实现资源发现、定位是

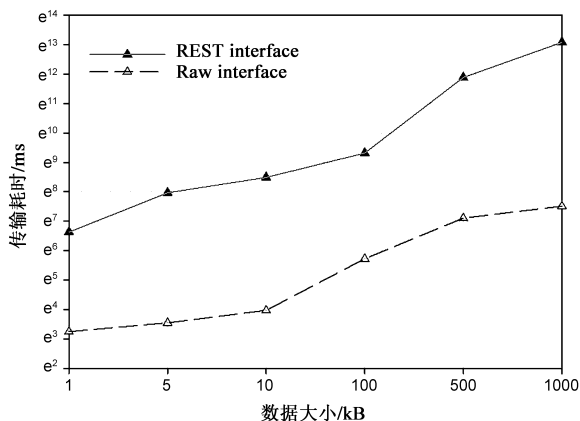


图6 REST接口与原生数据接口性能对比

个难题,本文基于 NIOT 物联网标识平台^[17-19]对海量资源进行统一的标识管理。

4.2.1 资源发现定位

实现应用间互联互通的关键是资源发现与定位,以确定数据所在位置。发现是指对单个物体对象查找其全生命周期所有资源,涉及的多个环节均可能建立

独立信息服务系统(如商品全生命周期的生产、物流、销售等环节),因此发现会涉及不同的信息服务系统;定位是确定单个资源的信息系统地址,通过解析服务实现。

本文采用 NIOT 管理资源标识,实现资源发现与定位。NIOT 平台基于 DNS 为物联网资源提供公共服务,包括资源注册、解析和搜索三个子系统。其系统架构及信息服务系统关系如图 7 所示。

针对物联网中的异构标识,NIOT 为兼容异构多编码,设计两段式的标识结构,标准码 + 物品码,标准码识别编码格式,物品码标识实体对象,分别由标准码名字服务器 SNS 和物品码名字服务器 TNS 提供服务。

4.2.2 协调工作流程

一个资源在信息服务系统中被创建时,其标识在 NIOT 中注册。注册信息包括:资源标识 (URI)、物品码、标准码,以及数据抽象标准的描述性信息(如:领域)等。NIOT 对收到的资源信息进行 URI 映射处理,完成注册。用户可使用解析系统定位数据位置;使用搜索系统实现相关资源的检索与发现。

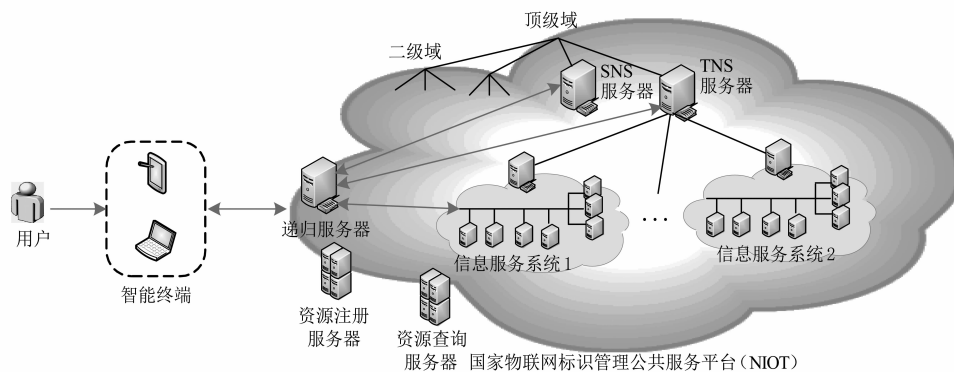


图7 信息服务系统与国家物联网标识管理公共服务平台

5 系统实现与评价

本文实现了原型系统,流程图如图 8 所示。系统采用 Java 语言,基于 JDK1.6 平台开发;使用 Oracle 的 Jersey 框架实现 REST 风格数据接口;结合 NIOT 实现数据定位;底层数据库采用 MongoDB;同时开发了中间件以实现数据的预处理。

下面从定性和定量两方面对原型系统进行评价。其中,定性分析主要分析开放性、可扩展性、灵活性、可靠性、高效性、可用性和安全性七个方面,定量分析则重点分析系统并发读写性能。

5.1 定性分析

本文选取三种物联网信息服务系统,包括 EPCIS^[20]、SeaCloudDM^[21]和 Sensor. Network^[12]。表 3 展示了三种方案与本方案的功能分析。

表 3 方案定性分析

	本方案	EPCIS	SeaCloudDM	Sensor. Network
数据接口	RESTful	SOAP	RAW	RESTful
数据格式	BSON/JSON	XML	CSV	JSON, XML, CSV
数据模型	Key-Value	RDBMS	RDBMS/ Key-Value	RDBMS
访问控制	fine-grained	fine-grained	fine-grained	user, group
认证方式	key, OAuth, pwd	N/A	N/A	pwd, key
兼容数据类型	any	textual, numeric	any	textual, numeric

在开放性方面,本文基于 REST 和 NIOT,数据共享机制贯穿数据抽象、发布、发现、定位和获取全流程,增

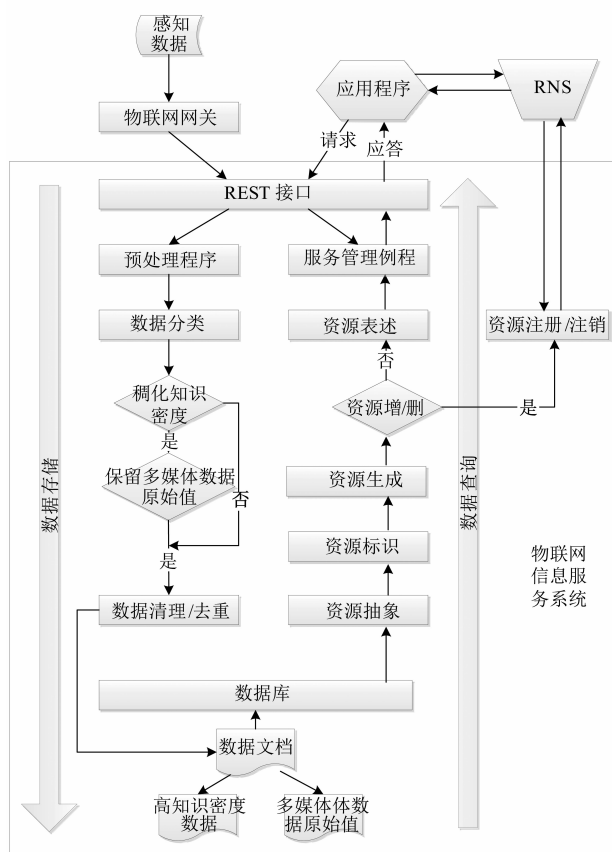


图8 系统流程图

强了其开放性,方便了数据共享.其他系统虽也采用REST,但很少考虑异构数据存储、表达、资源发现与获取等;另一方面,大多数REST应用只考虑系统功能,不考虑系统性能,本方案对资源表述形式和标准接口对系统整体性能的影响进行了评测与讨论,充分考虑到了性能方面的影响.

在扩展性和可靠性方面,现有方案大多采用RDBMS技术,可扩展性差.虽然有研究考虑使用非关系型数据库技术,但没有系统考虑存储策略;本方案的存储技术既可以横向扩展支持存储能力扩容,又可通过数据分片与备份机制,实现数据冗余存储,有效防止了单点失效带来的数据丢失.

在灵活性方面,一方面,感知数据的异构性、多属性特征导致其存储管理方式有很大不同,本方案的统一表达方式将不同类型感知数据转化为数值型及字符型存储,在面对异构数据有较高的灵活性;另一方面,利用NoSQL的存储特性,可轻松实现异构数据存储时数据字段的灵活增删.虽然SeaCloudDM在RDBMS基础上也提供基于键值形式存储,其性能很差.

在高效性方面,大多方案基于RDBMS,在海量数据下很难提供高性能查询,且并发支持度低;本方案基于NoSQL技术提供高并发能力,且设计了相关的存储策

略,以满足物联网数据实际需求,并进一步的提高数据质量,增强海量数据读写性能.

在可用性方面,本方案提供了标准的数据接口,减少了应用开发成本,提升了系统可用性,但服务功能还较少.SQL已不适应物联网时空相关性及动态流式等特征,而NoSQL也仅能支持功能有限的键值查询,所有方案都缺乏复杂、灵活的查询服务.

在安全性方面,由于物联网感知数据多涉及敏感、隐私信息,其安全问题需要专门考虑,本方案并未做相关研究,只是通过数据库的原生认证方式与访问控制方式一定程度地保证数据安全.

综上,本方案优点如下:支持海量数据,并通过分布策略保证数据平衡;标准化数据接口,且与NIOT结合,支持多平台数据共享;统一数据表达方式,便于管理;数据独立,支持多源异构的数据.

缺点包括:可用性有待加强,缺乏复杂、灵活的查询服务;数据安全保障有限.

5.2 定量分析

在性能上,从并发读和并发写两方面展开评测.

并发写测试分别使用1、2、4、8、16个线程向数据库插入数据,每个线程共插入10万条数据,每次写入一条;并发读测试分别使用1、2、4、8、16个线程对各个数据库执行查询,查询操作,数据规模约5000万条,结果集约5万条.

所有评测方案均采用REST风格接口,但本方案使用NoSQL技术,并采用了前述提到的存储策略,而其他方案采用RDBMS存储系统.

测试环境参数包括: Intel 至强 X5670 2.93GHz (6CPU) 的 CPU, 8GB 内存, 160GB 硬盘, 红帽 Linux 4.1.2-48 的 OS, mysql-5.5.17/mongodb-2.0.4 数据库, 测试语言为 Python/JAVA, 网络平均延迟 1.1ms. 本文实现了两个版本,早期系统采用 MySQL, 部署了 1 台服务器;而改进方案基于 MongoDB, 数据库集群包括 3 台服务器.

本次测评数据选取物联网真实数据,包括英特尔伯克利实验室 230 万条 WSN 数据^[22]和华盛顿大学计算机学院 lahar 项目约 330 万条 RFID 数据^[23].

性能评测结果如图 9 所示,在并发写情况下, MongoDB 方案可保持至少 20%, 最高约 200% 的性能优势;在并发查询情况下, MongoDB 方案时延远低于 MySQL 方案,且随并发数线性增长. 测试结果表明改进方案性能有明显提升.

另一方面,整体系统的测试较单纯数据库系统对比差异有所减少,这是因为:应用 REST 构建的标准接口导致了性能下降;采用集群方式存储海量数据,分片增加了数据的传输时延.

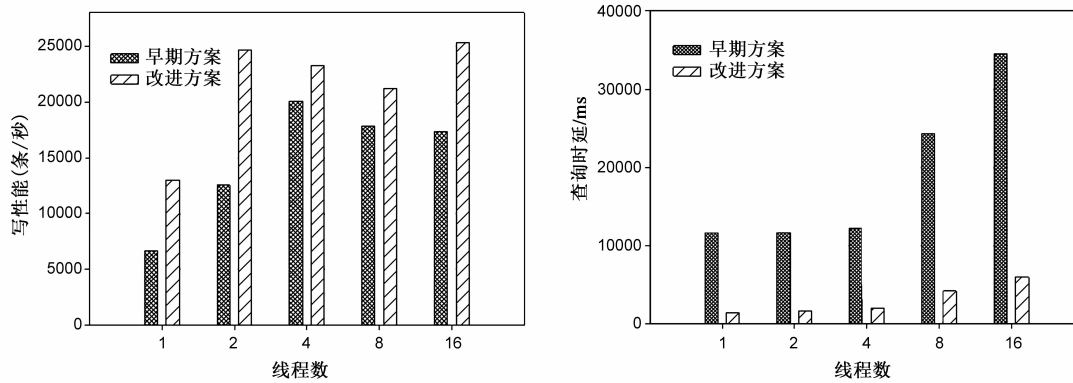


图9 方案性能对比

6 总结

本文针对物联网信息服务系统面对海量异构数据性能低下、共享困难等问题,开展了物联网信息服务系统存储与共享策略的研究,提出一种优化方案:(1)基于 NoSQL 存储技术提供高存储性能,并提出相应的预处理、数据分类以及数据分布策略,以提高数据质量,减少存储空间开销;(2)标准化的信息服务接口,且与国家物联网标识管理公共服务平台结合,支持多平台数据共享;(3)设计了统一数据表达方式,便于管理、理解与共享;(4)数据独立,采取的数据模型并未对感知的数据格式进行任何前提假设,可以灵活应对多源异构的数据。

本文主要采用 NoSQL 数据库解决数据存储共享问题,然而,它却缺乏关系数据库快速解决复杂问题的能力。下一步,将重点研究如何结合两种关系数据库的优点,进一步提高本文提出方案的性能。

参考文献

- [1] Li Tingli, Liu Yang, Tian Ye, et al. A storage solution for massive IoT data based on NoSQL [A]. Proceedings of the 2012 IEEE International Conference on Internet of Things [C]. USA: IEEE, 2012. 50 – 57.
- [2] Zhang Yin, Han Weili, Wang Wei, et al. Optimizing the storage of massive electronic pedigrees in HDFS [A]. Proceedings of the 3rd International Conference on the Internet of Things [C]. USA: IEEE, 2012. 68 – 75.
- [3] Zhang Guigang, Li Chao, Zhang Yong, et al. SemanMedical: a kind of semantic medical monitoring system model based on the IoT sensors [A]. Proceedings of the IEEE 14th International Conference on e-Health Networking, Applications and Services [C]. USA: IEEE, 2012. 238 – 243.
- [4] L Paul, M Dirk, B Andre. HashFS: applying hashing to optimize file systems for small file reads [A]. Proceedings of the International Workshop on Storage Network Architec-

ture and Parallel I/Os [C]. USA: IEEE, 2010. 33 – 42.

- [5] Zhang Yang and Liu Dan. Improving the efficiency of storing for small files in HDFS [A]. Proceedings of the International Conference on Computer Science & Service System [C]. USA: IEEE, 2012. 2239 – 2242.
- [6] Yang Hui, Qin Yong, Feng Gefei, et al. Online monitoring of geological CO₂ storage and leakage based on wireless sensor networks [J]. IEEE Sensors Journal, 2013, 13 (2) : 556 – 562.
- [7] Chang Penho, Wand Tsanpin. Supporting personal mobility with integrated RFID in VoIP systems [A]. Proceedings of the International Conference on New Trends in Information and Service Science [C]. USA: IEEE, 2009. 1353 – 1359.
- [8] H Gonzalez, Han Jiawei, Li Xiaolei, et al. Warehousing and analyzing massive RFID data sets [A]. Proceedings of the 22nd International Conference on Data Engineering [C]. USA: IEEE, 2006. 83.
- [9] 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.
Ding Zhiming, Gao Xu. A database cluster system framework for managing massive sensor sampling data in the Internet of things [J]. Chinese Journal of Computers, 2012, 35 (6) : 2514 – 2517. (in Chinese)
- [10] M D Francesco, Li Na, M Raj, et al. A storage infrastructure for heterogeneous and multimedia data in the Internet of things [A]. Proceedings of the 2012 IEEE International Conference on Internet of Things [C]. USA: IEEE, 2012. 26 – 33.
- [11] A M Maria, H L Sergio, S Abel, et al. A comparative study of data storage and processing architectures for the smart grid [A]. Proceedings of the First IEEE International Conference on Smart Grid Communications [C]. USA: IEEE, 2010. 285 – 290.
- [12] Deng Zhongliang, Xu Binxu, Li Ning. A sharing platform based on the Internet of things [A]. Proceedings of the In-

- ternational Conference on Computational Intelligence and Software Engineering [C]. USA; IEEE, 2010. 1 - 4.
- [13] IETF RFC 7252. Constrained application protocol (Co-AP), 2014 [S].
- [14] Gao Lei, Zhang Chunhong, Sun Li. RESTful web of things API in sharing sensor data [A]. Proceedings of the International Conference on Internet Technology and Applications [C]. USA; IEEE, 2011. 1 - 4.
- [15] D Guinard, M Mueller, V Trifa. RESTifying Real-world Systems a Practical Case Study in RFID [M]. Germany: Springer-Verlag, 2011. 359 - 379.
- [16] Liu Yimeng, Wang Yizhi and Jin Yi. Research on the improvement of MongoDB auto-sharding in cloud environment [A]. Proceedings of the 7th International Conference on the Computer Science & Education [C]. USA; IEEE, 2012. 851 - 854.
- [17] Tian Ye, Liu Yang, Yan Zhiwei, et al. RNS-a publice resource name service platform for the IoT [A]. Proceedings of the 2012 IEEE International Conference on Internet of Things [C]. USA; IEEE, 2012. 234 - 239.
- [18] 刘阳, 李馨迟, 田野, 等. 物联网名字服务关键技术研究 [J]. 电子学报, 2014, 42 (10): 2032 - 2039.
Liu Yang, Li Xinchu, Tian Ye, et al. Research on key technology of name service for the Internet of things [J]. Acta Electronica Sinica, 2014, 42 (10): 2032 - 2039. (in Chinese)
- [19] NIOT. 国家物联网标识管理公共服务平台 [OL]. <http://www.cniotroot.cn>, 2015.
- [20] GS1 Standard Version 1.1. EPC information services (EPCIS) version 1.1, 2014 [S].
- [21] Fan Chunxiao, Song Jie, Wen Zhigang, et al. A scalable Internet of things lean data provision architecture based on ontology [A]. Proceedings of the IEEE GCC Conference and Exhibition [C]. USA; IEEE, 2011. 553 - 556.
- [22] Intel Berkeley Research Lab. Intel lab data. [OL]. <http://db.csail.mit.edu/labdata/labdata.html>, 2004
- [23] University of Washington, CSE. The lahar project RFID data [OL]. <http://lahar.cs.washington.edu/displayPage.php?path=./content/Download/RFIDData/rfidData.html>, 2012.

作者简介



田野 男, 1979 年出生, 重庆涪陵人. 博士, 副研究员, 硕士生导师, 中国计算机学会高级会员. 2006 年博士毕业于中国科学院计算技术研究所, 2009 年进入中国科学院计算机网络信息中心. 从事物联网、下一代互联网、网络安全方面的有关研究.

E-mail: tianye@cnic.cn



袁博(通信作者) 男, 1984 年出生, 河北唐山人. 2013 年博士毕业于清华大学, 同年加入中国科学院计算机网络信息中心. 主要从事物联网标识方面的研究工作.

E-mail: yuanbo@cnic.cn

李廷力 男, 1987 年出生, 四川什邡人. 2013 年硕士毕业于中国科学院计算机网络信息中心. 从事物联网信息处理方面的研究工作.