

基于消息中间件及 MongoDB 的物联网应用服务平台^①



刘峰^{1,2}, 王昭鹏^{1,2}, 于波^{1,2}, 郭放达³

¹(中国科学院大学, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110168)

³(东北大学 计算机科学与工程学院, 沈阳 110819)

通讯作者: 王昭鹏, E-mail: 1062576512@qq.com

摘要: 物联网正广泛应用于各行各业. 将现实世界参数化并结合感知设备、网络通信技术、数据技术等诸多手段实现物联网数据对于用户的直接可用便是物联网技术的中心思想. 本文依托于智能燃气数据管理系统, 针对其物联网平台架构中应用服务平台在大规模设备接入场景下的性能瓶颈, 提出一种结合消息中间件 Kafka 与 NoSQL 数据库 MongoDB 的混合方案. 根据燃气公司和设备厂商的应用背景实现该方案, 完成应用服务平台并发性能和数据持久化效率的提高.

关键词: 物联网; 应用服务平台; 消息中间件; MongoDB

引用格式: 刘峰, 王昭鹏, 于波, 郭放达. 基于消息中间件及 MongoDB 的物联网应用服务平台. 计算机系统应用, 2019, 28(5): 90-94. <http://www.c-s-a.org.cn/1003-3254/6891.html>

IoT Application Service Platform Based on Message Middleware and MongoDB

LIU Feng^{1,2}, WANG Zhao-Peng^{1,2}, YU Bo^{1,2}, GUO Fang-Da³

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

³(School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China)

Abstract: The Internet of Things (IoT) is widely used in all walks of life. The central idea of IoT is to parameterize the real world and combine with sensing devices, network communication technique, data technologies, and other means to realize the direct usability of the IoT data for users. Based on the Shancheng intelligent gas data management system, this study proposes a hybrid scheme combining the message middleware Kafka and the NoSQL database MongoDB for performance bottleneck of the application service platform in the IoT platform architecture when encounters large scale device access scenarios. The scheme is implemented based on the application background of gas companies and equipment manufacturers, the application service platform concurrency performance and data persistence efficiency are improved.

Key words: Internet of Things (IoT); application service platform; messaging middleware; MongoDB

当今物联网 (Internet of Things, IoT) 技术俨然已经成为蓬勃发展的技术体系^[1]. 其应用领域广泛, 涉及工业、农业、交通、城市建设等诸多领域的诸多场景,

因此物联网平台建设成为业界热门课题.

世界各研究组织对物联网架构划分相似又不尽相同, 在 ICT 中国. 2010 高层论坛上中国电信上海研究

① 收稿时间: 2018-11-23; 修改时间: 2018-12-12; 采用时间: 2018-12-17; csa 在线出版时间: 2019-05-01

院李安民院长指出物联网架构划分为感知层、网络层、应用层三层. 目前国内外知名物联网平台虽根据各自的技术侧重对整体架构层次有差异地进行详细划分, 但仍可认为上述三层划分是具有普适性的基本划分.

本文依托于智慧城市项目“智能燃气数据管理系统”^[2], 该项目基于 CoAP 协议实现燃气设备智能管控.

全文首先介绍消息中间件 Kafka 和 MongoDB 数据库, 然后以项目为例, 结合物联网平台特点指出架构中存在的性能瓶颈. 并针对发现的问题提出一种基于消息中间件 Kafka 和 NoSQL 数据库 MongoDB 的应用服务平台架构. 最后进行新旧架构数据读写性能比较和并发稳定性测试, 验证新方案的可行性和有效性.

1 技术背景

1.1 消息中间件 Kafka

在大量数据传递场景中, 消息中间件能够使传递双方单单关注数据而非如何共享, 并通过维护消息队列实现异步工作方式, 因此消息中间件往往被简单看作为消息队列. 其主要有两种异步消息模式: 点对点模式和发布-订阅 (Pub-Sub) 模式.

Kafka 是 Apache 软件基金会开源项目, 专为高并发场景而设计的分布式消息中间件^[3], 它采用基于消息主题 (Topic) 的发布-订阅消息模式, 支持不同消费者分布式消息读取. 与传统消息队列相比, Kafka 单点生产者消费者达到每秒百万级读写, 具有更高的吞吐量; 它将数据写入磁盘并复制到集群中保证数据持久化和容错性.

1.2 NoSQL 数据库 MongoDB

NoSQL (Not Only SQL) 数据库即非关系型数据库, 从数据模型角度分为四类: 键值 (Key-Value) 型、文档 (Document) 型、列式 (Column Family)、图式 (Graph)^[4]. MongoDB 是文档型数据库的典型代表, 兼具关系型数据库和非关系型数据库的部分特点和功能, 它不需要预先定义表结构, 其数据结构较为松散灵活, 支持内嵌文档等复杂结构, 适合物联网平台中海量半结构化和非结构化异构数据的存储^[5].

2 方案研究

2.1 现有方案简介

根据物联网平台架构整体划分, 现今普遍采用的整体架构方案如图 1 所示.

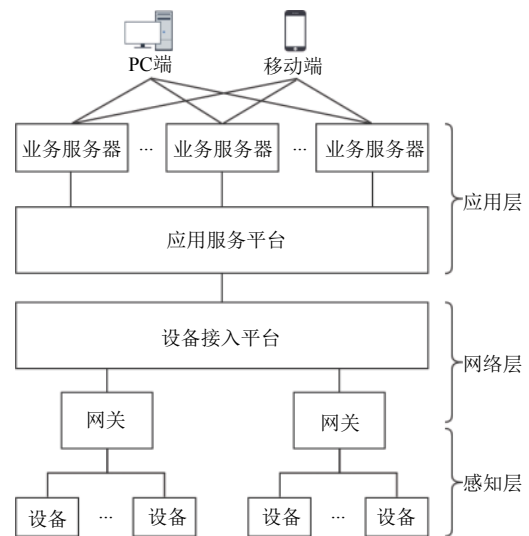


图 1 物联网平台整体架构

设备接入平台支持物联网终端设备使用标准物联网协议 (CoAP, MQTT 等) 完成适配接入, 通过协议解析提取数据, 最后向应用服务平台进行交付. 应用服务平台在整体架构中具有承上启下的作用, 一方面基于应用服务器集群的存储能力和计算能力对设备接入平台提交的数据进行持久化和数据处理, 另一方面面向业务服务器提供双向接口供第三方调用. 业务层主要通过 B/S Web 应用面向各行各业, 实现设备信息展示、设备控制操作、数据统计展示等功能.

2.2 基于消息中间件及 MongoDB 的应用服务平台方案

大量理论研究^[6-8]和工程实践经验总结出物联网平台具有如下特点: 第一数据量大、并发性高, 据调研现今大型商用物联网平台需求支持至少千万级设备长连接接入和百万级并发能力; 第二数据多源异构、结构松散, 该特点主要原因是物联网设备的多样性, 从而产生大量异构数据无法事先统一结构.

通过分析现有方案发现两处与上述特点不相适应之处: 第一在大规模设备接入场景中, 设备接入层消息代理服务器有限的消息交付能力和沉重的工作量导致出现性能瓶颈. 第二传统关系型数据库 MySQL 其固定的表结构及字段属性无法较好地适应异构松散的设备数据, 同时严格的事务约束也限制了读写效率.

针对上述问题, 提出一种基于消息中间件 Kafka 和 MongoDB 数据库的应用服务平台方案, 改变与设备接入平台原有的对接方式, 改变数据持久化方式. 该方案中采用性能更优的扩展钩子 (Hook) 将接入平台中相

对分散的消息数据汇聚至 Kafka. 应用服务器集群中各服务器节点分布式消费 Kafka 中的消息, 从而实现一次数据分区, 并在完成相应处理后以分片分页策略持久化至 MongoDB.

3 应用服务平台设计实现

基于本文提出的方案, 应用服务平台架构如图 2 灰色部分所示

本方案采用扩展钩子与 Kafka 相结合的方式完成接入平台与应用服务平台对接, 两者较好的并发性能

会提高数据流转效率. 同时所有数据统一汇集至 Kafka 进行缓冲, 一方面可取消接入平台对消息主题路由表的使用和维护, 减少消息转发的工作量. 另一方面可对接入平台数据进行容灾保障. 利用 MongoDB 数据库集合 (Collection) 中文档数据结构的灵活性, 对半结构化数据直接存储, 避免繁琐的数据结构转换, 化简数据持久化过程. 此外 MongoDB 提供的聚合方法 aggregate() 可以对集合中的文档进行变换组合, 而且支持 Map-Reduce 计算模式, 足以应对业务平台对大数据分析及海量数据高并发操作的应用需求.

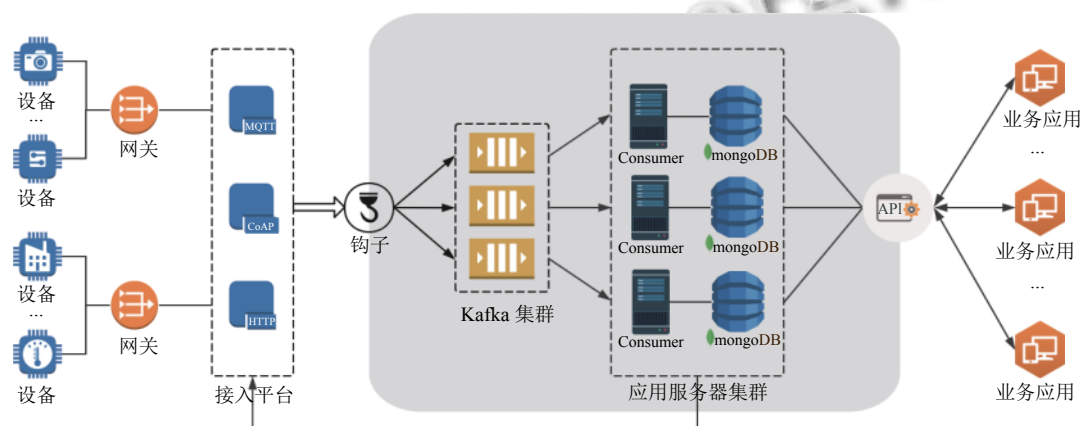


图 2 应用服务平台架构

3.1 扩展钩子设计实现

扩展钩子负责设备接入平台与应用服务平台对接, 本质是以生产者身份将接入平台消息按照主题发布至 Kafka 集群. 扩展钩子采用职责链设计模式, 使用 Erlang 语言实现^[9], 通过向钩子注册回调函数, 在接入平台接收设备终端上下线、状态参数更新时触发钩子并顺序执行回调函数. 事件触发与回调函数对应关系如表 1 所示.

表 1 扩展钩子对照表

钩子注册名	回调函数
ClientConnect	on_client_connected
ClientSubscribe	on_client_subscribe
ClientUnsubscribe	on_client_unsubscribe
ClientDisconnect	on_client_disconnected
MessagePublish	on_msg_publish
MessageDelivered	on_msg_delivered
MessageAcked	on_msg_acked

3.2 数据持久化方案设计与实现

本方案主要从数据格式、数据存储模式、数据操作三方面切入:

数据格式选用 JSON 格式, 因其 Key-Value 文本结构与 MongoDB 的 BSON 存储格式极为类似, 便于发挥 MongoDB 松散数据结构优势进行设备数据存储^[10], 同时易于设备生成、解析, 开发人员阅读、编写.

数据存储选用 MongoDB 数据库, 根据 MongoDB 官方性能测试报告显示, 在读写平衡场景中, MongoDB 可达 135 000 ops/sec. 在此基础上, 面向设备接入平台采用 Kafka 与 MongoDB 相结合方式, 旨在减轻大规模数据写入时数据库工作负荷. 首先设备数据汇聚至 Kafka 缓冲, 避免直接对数据库造成冲击. 其次利用 Kafka 消息队列特性, 在生产者-消费者模式中应用服务器集群各节点以消费者身份消费特定主题下消息数据, 实现一次数据分区, 完成数据在特定节点持久化. 主题可根据具体项目应用场景灵活设置, 例如根据地理区划、行业领域等设置主题. 最后利用 MongoDB 支持自定义存储过程的特性, 集合采用逻辑上的双端队列, 基于最近最久未使用 (LRU) 算法, 使用 JavaScript 实现数据分页存储相应的增删改查方法. 以查找为例,

按页号遍历文档队列,找到特定数据页后 pageIndex(当前页号)不变,修改字段 lastIndex(前页页号),nextIndex(后页页号)值,使之位于队首,示意图如图3所示.

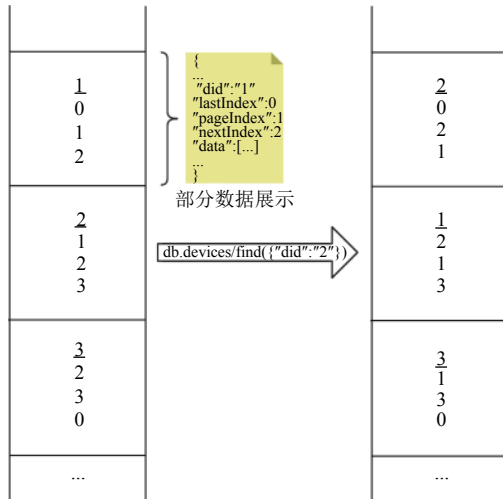


图3 双端队列 LRU 示意图

数据操作主要面向业务层,采用 REST 架构风格的 Web 服务封装简单易用的接口并设计实现相应 URL 定位,通过 HTTP 请求完成数据的增删改查操作.

3.3 应用服务平台数据流转过程

本文设计实现的应用服务平台数据流转过程主要

包括数据采集汇聚、数据处理、数据分区、数据持久化.扩展钩子通过事件触发机制完成数据采集;Kafka 采用生产者消费者模式完成数据汇聚和数据分区,同时可对数据进行处理;MongoDB 基于文档模型采用分页方式完成数据持久化,为业务层提供可用数据;整体数据流转时序如图4所示.

4 测试

测试机配置: Intel(R) Core(TM)i5-4200U CPU @ 1.60 GHz, RAM 8 GB DDR3, 硬盘 240 GB SATA, 系统 CentOS7.

本文基于消息中间件及 MongoDB,针对原有物联网平台架构中应用服务平台进行改进,并结合智能燃气数据管理系统具体性能需求完成搭建部署.系统运行如图5所示.

为验证本文方案的有效性,通过使用 Web Bench 压测工具进行百万级数据读写性能对比测试,结果如图6所示.使用 emqtt_benchmark 工具进行平台稳定性测试,结果如表2所示.以上结果表明本方案中数据库持久化效率虽有波动但得到明显提升;在当前测试条件下平台服务器节点 CPU 利用率收敛于 59%,内存占用处于合理范围,平台吞吐量总体趋于平稳,可见平台具有较好的性能稳定性.

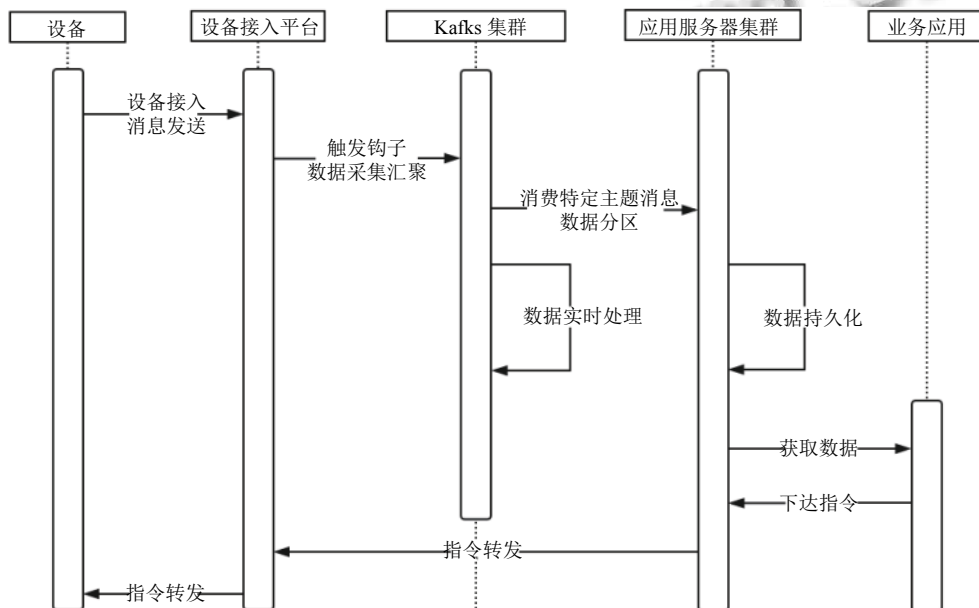


图4 数据流转时序图

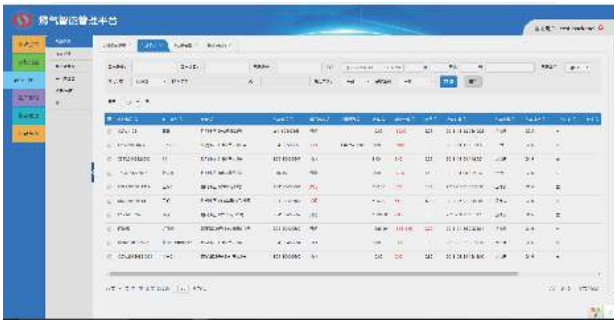


图5 系统运行效果

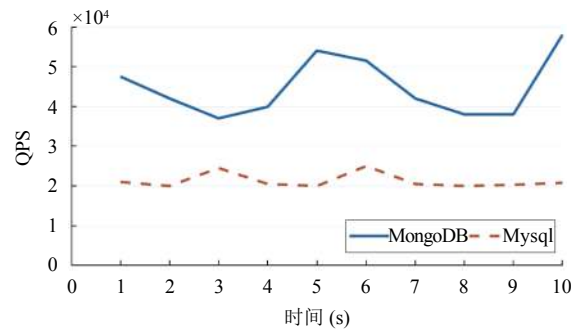


图6 数据库读写性能对比

表2 平台稳定性测试

模拟设备数	消息发布速率 (msg/s)	时间 (ms)	已接收消息数	CPU 利用率	内存	平台吞吐量 (msg/s)
100	100	526 001	5 259 671	57%~64%	≤2.74G/8G	9 999
		527 008	5 269 674			9 933
		528 002	5 279 673			10 057
		529 000	5 289 668			10 017
...

5 结论

本文依据物联网平台特点,分析常见物联网平台架构性能瓶颈成因,针对应用服务层提出一种基于消息中间件 Kafka 及 NoSQL 数据库 MongoDB 的应用服务平台方案,设计并实现了扩展钩子和数据持久化模式,改变了接入平台与应用服务平台之间的对接方式,改变了数据流转方式和持久化方式.实验结果证明,本文方案切实有效,平台并发性能稳定、持久化效率得到提高,并且后续开发维护难度降低,更好地支持业务扩展.

后期工作将充分利用 Kafka 在数据流实时处理过程中的优势,进一步提升平台性能并为业务层提供更多数据支持.

参考文献

- Conti JP. The internet of things. *Communications Engineer*, 2006, 4(6): 20–25. [doi: 10.1049/ce:20060603]
- 孙建伟, 邹慧丽, 于碧辉. 基于 CoAP 的智能燃气数据管理系统. *计算机系统应用*, 2018, 27(8): 103–107. [doi: 10.15888/j.cnki.csa.006470]

- Kafka A. A high-throughput distributed messaging system. kafka.apache.org.
- 唐洪奎, 张程, 刘骥. 基于 NoSQL 的物联网数据本体模型存储技术研究及实现. *软件*, 2017, 38(3): 27–33. [doi: 10.3969/j.issn.1000-386x.2017.03.005]
- Chodorow K, Dirolf M. *MongoDB 权威指南*. 程显峰, 译. 北京: 人民邮电出版社, 2011.
- Vladislavs A, Eduards F, Jānis K. Internet of things: Structure, features and management. *Information Technology and Management Science*, 2016, 19(1): 78–84.
- 张桂刚, 毕娅, 李超, 等. 海量物联网数据安全处理模型研究. *小型微型计算机系统*, 2013, 34(9): 2090–2094. [doi: 10.3969/j.issn.1000-1220.2013.09.025]
- Alkhalil A, Ramadan RA. IoT data provenance implementation challenges. *Procedia Computer Science*, 2017, 109: 1134–1139. [doi: 10.1016/j.procs.2017.05.436]
- Kruger K, Basson A. Erlang-based holonic controller for a palletized conveyor material handling system. *Computers in Industry*, 2018, 101: 120–126. [doi: 10.1016/j.compind.2018.07.003]
- 吕林. 基于 MongoDB 的应用平台的研究与实现[硕士学位论文]. 北京: 北京邮电大学, 2015.