

文章编号: 2095-2163(2021)05-0001-05

中图分类号: TP 274+.2

文献标志码: A

# OPCUA 网关数据存储系统

王灿达, 李悦, 李锋

(东华大学 计算机科学与技术学院, 上海 201620)

**摘要:** OPCUA 网关对外部系统提供统一标准的数据接口,是智能制造应用领域的关键产品。针对 OPCUA 网关的数据存储问题,设计了一套内存与外存协同的数据存储系统。摒弃商业数据库软件,直接基于内存与外存进行数据操作。定义了应用于 OPCUA 结点数据的内存存储结构与外存存储结构,并在此结构的基础上设计了网关数据写入与读取的流程。该系统在 Raspberry Pi 3B 硬件平台上实现,并与成熟的数据库系统进行对比测评。实验结果表明,该系统在数据的读、写以及硬件资源使用率等方面均有较好的表现。

**关键词:** 智能制造; 边缘计算; 数据网关; OPCUA

## OPCUA gateway data storage system

WANG Canda, LI Yue, LI Feng

(College of Computer Science and Technology, Donghua University, Shanghai 201620, China)

**[Abstract]** The gateway based on OPCUA provides a unified standard data interface for external systems, which is a key product in the application field of intelligent manufacturing. Aiming at the data storage problem of OPCUA gateway, a data storage system based on random access memory and external storage was designed. The system abandons the commercial database software and operates data directly based on random access memory and external storage. The random access memory storage structure and external storage structure which are applied to OPCUA node data were designed, which is foundation of the process of gateway data writing and reading. The system was implemented on raspberry PI 3B hardware platform and was compared with the mature database. The experimental results showed that the system had good performance in data writing, reading and hardware resource utilization.

**[Key words]** intelligent manufacturing; edge computing; data gateway; OPCUA

## 0 引言

伴随着《中国制造 2025》战略计划的不断推进,学术界与工业界将越来越多的目光投向了智能制造的研究与实践。为了生产数据实时处理的需要,数据在边缘侧处理后,再与云计算中心进行交互,这种云边协同的模式是智能制造发展的一个新趋势<sup>[1]</sup>。保证生产信息的收集、存储与传输,使得云中心能快速方便地获取来自于生产设备的信息,是实现云边协同的关键因素。但目前生产设备种类繁多,通讯接口协议各异,上层应用系统需针对各类设备单独开发子模块用于数据交互,这无疑增大了开发的难度,阻碍了智能制造的发展<sup>[2]</sup>。OPC 统一架构即 OPCUA,定义了一组高可用、高性能和跨平台的数据交互规范<sup>[3]</sup>。基于 OPCUA 技术所开发的网关可屏蔽不同设备的通讯细节,对外部系统访问生产设

备数据提供统一标准的接口<sup>[4]</sup>。在实际生产中,网关处于生产设备边缘,由于现场网络的不确定性较高,通讯不稳定,存在数据丢失的风险。而在工业生产中,网关所传输的数据涉及到设备主要生产指标以及控制数据,对数据的完整性要求较高。因此,OPCUA 网关中应具备数据存储能力,遇到网络中断时,网关可继续准确采集,并将数据存储到非易失设备中,当网络恢复正常时,其它第三方平台可从网关中获取完整数据。这就保证了可靠的数据服务,提高了系统的鲁棒性与容错性<sup>[5]</sup>。

目前,在智能制造网关的研究中,学者大多关注于不同协议之间的转换<sup>[6]</sup>、信息通讯的安全性<sup>[7]</sup>以及网关的边缘计算能力<sup>[8]</sup>等方面,部分学者注意到了网关对于数据的存储能力,但现有的数据库系统仍是其不二选择。文献[9]在对基于 OPCUA 的工控信息化数据网关的研究中采用了 SQLite;文献

**基金项目:** 工信部智能制造新模式——间歇式智能印染工厂。

**作者简介:** 王灿达(1996-),男,硕士研究生,主要研究方向:智能制造;李悦(1981-),男,博士,讲师,主要研究方向:信息安全、密码学、数据分析。

**通讯作者:** 李悦 Email: frankyueli@dhu.edu.cn

**收稿日期:** 2021-01-05

[10]在设计家庭看护系统的智能网关时选择了 MongoDB。此外 MySQL、HBase 也广受学者们的青睐<sup>[11]</sup>。由于数据库的运行需要较多的系统资源,这就对硬件配置提出了较高的要求,而网关设备硬件资源紧张,运行数据库软件较为困难。此外,操作数据库需要建立网络连接、传输数据、关闭连接等一系列耗时步骤,这会导致网络 I/O 速度成为系统运行效率的瓶颈。现有的数据库软件为追求更广泛的应用场景,其存储与查询设计具有一定的通用性,并不会针对 OPCUA 网关程序做专门的优化。基于上述现状,本文设计了一套内存与外存协同的数据存储系统(Storage System Based On Ram And External Storage Data,简称 RES),专用于对 OPCUA 网关数据的存储。RES 只保留了核心的数据存储与查询功能,抹除了事务、存储过程等额外的数据库功能,故其资源占用率小,硬件要求低;另外 RES 可直接嵌入到 OPCUA 应用程序中,二者之间的互操作不再需要占用网络 I/O 资源,大幅度提升了程序运行效率。

## 1 OPCUA 网关总体架构

OPCUA 网关采集设备信息并存储,对外部呈现 OPCUA 数据接口。在逻辑上,网关系统分为 3 层,分别为采集层、存储层与服务层,如图 1 所示。采集层读取生产设备数据,整合目前采集生产数据常用的接口(如 RS485、RS232、USB 等),并针对不同的通讯方式开发不同的驱动程序完成数据的采集,之后将数据写入到存储层。存储层为服务层提供数据支撑,分为内存存储与外存存储两部分。外部系统通过 OPCUA 客户端与网关服务层建立安全连接。连接成功的客户端可浏览服务层地址空间,访问结点的当前数据与历史数据,还可以对节点进行订阅,当数据更新时自动获取。

## 2 内外存协同的数据存储

数据库可划分为二种,一种是传统的关系数据库,主要以外存作为存储介质,如 Oracle、MySQL 等,数据的访问会同时涉及磁盘 I/O 与网络 I/O,速度较慢;另一种是实时数据库,主要以内存作为存储介质,如 Redis、Memcached 等,此类数据库直接基于内存操作,速度较快,但受限于内存资源,其存储容量有限。若将二者结合,外存型数据库用于持久化存储“冷数据”,即访问频次较低的数据;内存型数据库缓存“热数据”,即访问频次较高的数据,可扬长避短提升系统性能。本文的 RES 就是基于上述思想所设计,用以

适应 OPCUA 网关中采样数据管理,在有限的硬件资源下可保证数据的读写速率与存储容量。

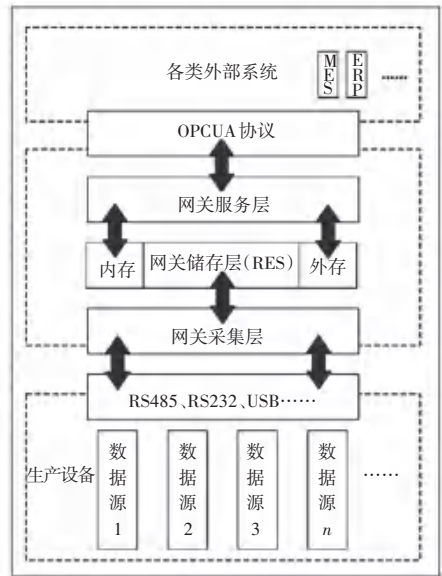


图 1 网关整体架构

Fig. 1 Gateway architecture

### 2.1 内存储结构设计

生产过程中设备数据具有数值和时间属性,本文采用 OPCUA 标准中的 DataValue 数据对象来表示。DataValue 的各属性见表 1。

表 1 DataValue 属性

Tab. 1 Datavalue property

名称	作用
value	在状态码指明读取成功的情况下,包含读到的值
statusCode	读取操作的状态码,若其为 Good,值才是可用的
sourceTimeStamp	表示数据源应用于节点的时间戳
serverTimeStamp	表示服务器收到变量值的时间

DataValue 对象可描述某个节点在某一时刻的具体值,对其按照时间顺序存储,则可记录某一节点数据随时间的变化情况。由于数组可满足对元素的有序存储,故本文为每个节点建立一个固定长度的 DataValue 数组,在内存中存储该节点的时序数据。为建立起节点与数组的对应关系,方便根据节点的唯一标识符(nodeId)对数组进行查找,本文引入哈希集合作为最外层容器。哈希集合以键值对的方式进行数据映射与存储,每个节点的 nodeId 作为键,与节点对应的数组作为值共同存储在哈希集合之中。内存存储结构如图 2 所示。

### 2.2 外存存储结构设计

外存存储结构三要素为:记录、数据页、索引,如图 3 所示。记录是节点数据在外存中最基本的存储形式,每行记录包含 3 个字段见表 2,分别是

nodeNumber、value、timeStamp。为方便对记录排序, 本文对 OPCUA 模型中所有节点重新编号并与其 nodeId 一一对应。

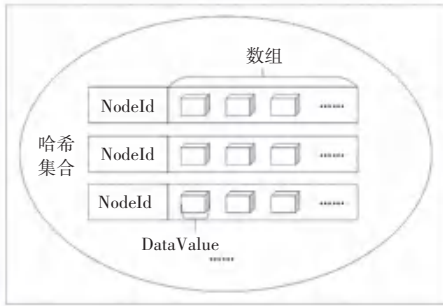


图 2 内存存储结构

Fig. 2 RAM storage structure



图 3 外存存储结构

Fig. 3 External storage structure

表 2 记录字段

Tab. 2 Record field

名称	说明
nodeNumber	节点编号, 与节点的 nodeId 一一对应
value	此次数据更新的具体值
timeStamp	表示服务器收到变量值的时间

数据页保存着记录, 可分为 3 部分: 头信息、页目录与记录。头信息是为生成索引服务的, 包含数据页所有记录中最早时间的时间戳与最晚时间的时间戳; 每个节点所对应的记录在此数据页的位置区间存放于页目录之中, 通过查阅页目录可高效地根

据节点编号锁定想要的数。为方便对数据页的组织管理, 数据页大小固定且以阿拉伯数字顺序从 1 开始逐个编号, 越晚生成的数据页其页号越大。索引页只记录数据页页号及此数据页的头信息, 在查询所需数据时遍历索引, 可快速定位目标数据页。

### 2.3 数据的写入与访问机制

#### 2.3.1 数据写入机制

设备数据从网关采集层写入网关存储层的流程如图 4 所示。首先会在 OPCUA 地址空间中更新此节点的当前值, 之后根据此值封装一个 DataValue 对象放入与该节点对应的数组, 用于在内存中缓存节点的“热数据”。若数据量超出了数组的容量, 则会按照时间顺序对早期放入数组的数据进行清除, 确保内存中总是缓存设备新产生的数据, 在实际生产中新数据被访问的频次大于旧数据。

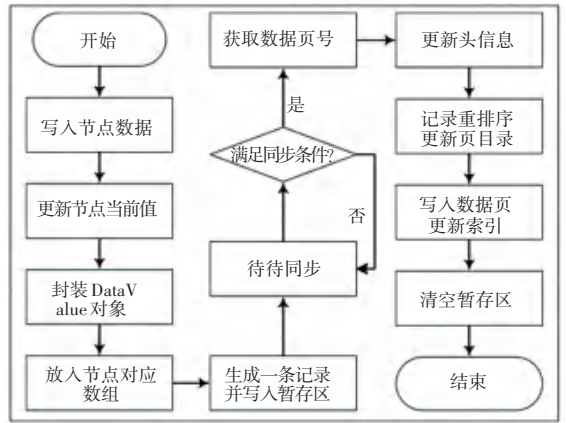


图 4 数据写入流程

Fig. 4 Data writing process

为确保节点数据的完整性以及整个系统的鲁棒性, 当数据更新时还会生成一条记录, 用于在外存中持久化存储。考虑到外存 I/O 耗时较长, 此记录并不会立即同步至外存, 而是先放入位于内存的记录暂存区, 等待时机批量写入, 以减少 I/O 次数, 从而提升系统性能。记录由暂存区写入外存的条件: 一是每隔固定的时间间隔进行一次数据同步, 二是暂存区数据量达到规定数目, 满足任意一个即可。

记录从暂存区写入外存过程如下:

(1) 获取待写入数据的数据页。计算最晚生成的即页号最大的数据页的长度, 若超过规定的长度则创建新的数据页, 否则直接写入到此数据页中。

(2) 更新数据页的头信息。取暂存区中最晚记录的时间戳来更新头信息的最晚时间。若头信息中已有最早时间则无需更新, 否则取暂存区最早记录的时间戳来设定头信息的最早时间。



(3)记录排序。将暂存区的记录与数据页中原有的记录混合,并按照编号与时间戳这 2 个字段进行升序排列。

(4)更新数据页的页目录。遍历排序后的记录值,得出每个节点所对应的记录的位置区间,最后生成新的页目录。

(5)将头信息、页目录、有序记录写入数据页,并在索引中更新此数据页的头信息。

(6)清空记录暂存区。

### 2.3.2 数据访问机制

各类外部系统通过网关服务层对数据进行访问分为两种情况:一是对于节点当前值的读取,这种方式比较简单直接依据 OPCUA 规范返回对应数据即可;另一种是对于节点在某段时间范围内历史数据的读取,此时需要查询 RES 以获取数据。3 个关键查询信息分别为 nodeId、开始时间、结束时间。详细步骤如下,流程如图 5 所示。

(1)根据 nodeId 在哈希集合中查找到存放 DataValue 对象的数组。

(2)遍历数组,若满足查询条件则返回对应数据,若不满足则查找位于外存的记录。

(3)根据开始时间与结束时间遍历索引,得到满足条件的若干个数据页。

(4)逐一处理数据页,由 nodeId 所对应的编号去查询页目录定位并获取数据。

(5)把此次在外存中查找到的数据放入内存对应的数组中。即“冷数据”升级为“热数据”,提升下次查询的速率。

(6)将在内存与外存中查找到的数据一并返回给客户端。

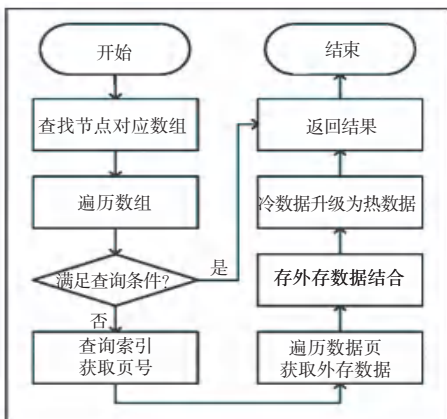


图 5 数据读取流程

Fig. 5 Data reading process

## 3 方案测试

本文以染整车间中定型机为基础搭建 OPCUA 网关,以对 RES 进行测试。定型机包含多个监控单元,每个监控单元监控某项指标。如,烘房温度、布面含潮率,同时也对应着 OPCUA 地址空间中的一个节点。网关采集层通过 RS485 接口从定型机的监控单元获取数据存入 RES,网关服务层接收 OPCUA 客户端的请求并返回相应数据。

网关的硬件平台基于 Raspberry Pi 3B,其 CPU 配置为 64 位 1.2 GHz 四核心,内存 1 G、外存 32 G。本文选取了以外存为存储介质的关系型数据库 MySQL 和以内存为存储介质的键值对数据库 Redis 作为 RES 的比较对象,来评测 RES 的各项性能指标。在 MySQL 中建立数据表存储节点数据,具体字段信息见表 3。为提升数据查询的速率,在该表中以 update\_time 与 node\_id 字段建立了索引。

表 3 MySQL 数据表信息

Tab. 3 MySQL data table information

字段名称	数据类型	是否为主键	是否为索引	用途
id	int	是	否	计数
node_id	varchar	否	是	对应 OPCUA 节点
update_time	timeStamp	否	是	记录数据更新时间
value	float	否	否	该节点对应的值

节点数据在 Redis 中以 hash 表的方式进行存储,每一个 OPCUA 节点对应于 Redis 中的一个 hash 表,即 Redis 中的每一个 hash 表都以 OPCUA 节点的 nodeId 当做键值。在 hash 表中该节点数据以更新时刻的时间戳作为字段名称,这一时刻节点的值当做与该字段对应的值,具体存储结构如图 6 所示。

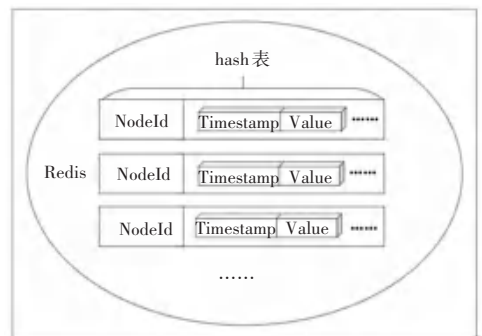


图 6 Redis 存储结构

Fig. 6 Redis storage structure

为测试不同存储方案下数据的写入速率,本文随机模拟定型机在运行时所产生的数据,以连续写入 1 000 条数据为起始,步长为 1 000,递增到

10 000 条。每个量级写入 5 次,并计算出平均写入时间,具体结果如图 7 所示。MySQL 写入数据所用时间最多,且随着数据量的增大,所用时间的增幅也非常明显;Redis 与 RES 写入速度较 MySQL 有明显的提升,数据量增大时,所用时间的增幅趋于平缓;Redis 的写入速率略高于 RES。在写入时 MySQL、Redis、RES 这 3 种方案 CPU 的平均使用率为 38%,32%,29%;内存平均使用率为 35%,28%,27%。

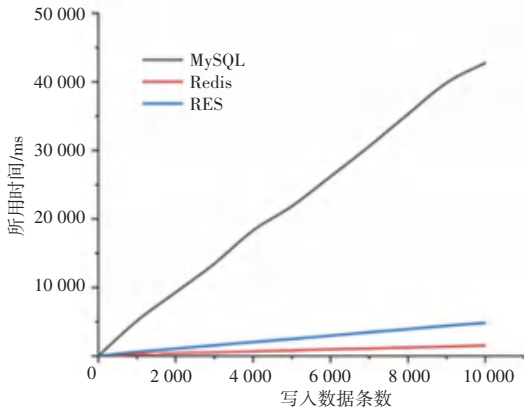


图 7 不同方案写入数据所用时间

Fig. 7 Data writing time of different schemes

对于数据读取速度的测试,本文首先在 MySQL、Redis、RES 中分别存入 10 000 条记录,并随机读取某个节点在某个时间段内的历史数据。统计读取 100 次,以 100 为步长递增到 1 000 次不同方案所用的时间,具体结果如图 8 所示。MySQL 对于 SQL 语句有着很好的支持,一条简单的 SQL 语句即可完成查询,虽操作方便但速度并不理想;Redis 对于数据的操作方式较少,且其内部并不会对数据按照时间排序,故只能采用遍历的方式筛选出合格的记录;RES 在写入时对数据建立了索引与页目录,并且数据全部按照时间顺序排列,使其在查询时无需遍历,可以直接查找到所需记录,大幅度提升了查询效率。在查询时 MySQL、Redis、RES 这 3 种方案 CPU 的平均使用率为 30%,29%,28%;内存平均使用率为 38%,45%,29%。

在实际工业生产中数据量巨大,在千万级别的数据量下 RES 仍有较好的表现。写入性能方面,MySQL 与 RES 写入 10 000 000 条数据分别用时约 30.91 h、2.24 h。由于网关硬件资源有限,内存只有 1 GB,当 Redis 的数据写入量达到 60 000 000 左右时,则已达到其容量瓶颈不能继续操作。数据查询方面,基于千万级数据量随机查询 100 次 MySQL、RES 所用时间分别为 9 282.89 s、1 722.16 s。

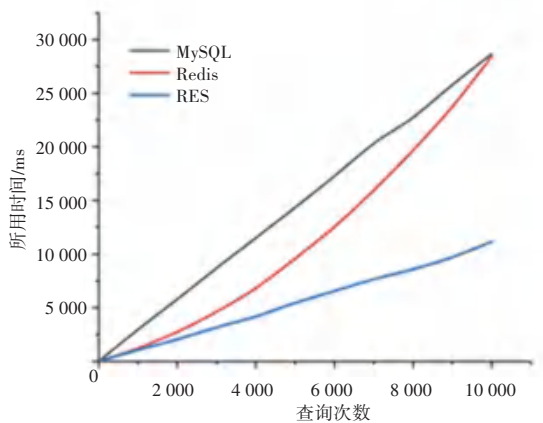


图 8 不同方案查询数据所用时间

Fig. 8 Data query time of different schemes

## 4 结束语

本文对网关中的数据存储进行了研究,设计并实现了内存与外存协同的数据存储系统,内存做数据缓存,外存做持久化存储,兼顾了数据存储容量与操作效率,在计算资源受限的开发板上,相比传统的数据库 MySQL 与 Redis 有更好的性能表现。OPCUA 网关解决了不同生产设备之间、生产设备与外部系统之间难以互联互通的问题,对于 OPCUA 网关的进一步发展具有一定的参考意义。

## 参考文献

- [1] PREMSANKAR G, FRANCESCO M D, TALEB T. Edge computing for the internet of things: A case study [J]. IEEE Internet of Things Journal, 2018, 5(2): 1275-1284.
- [2] 李锋,张坤,原丽娜. 基于 OPC UA 的纺织智能染整车间信息模型研究与实现[J]. 纺织学报,2020,41(2): 149-154.
- [3] POLGE J, ROBERT J, TRAON Y L. Assessing the impact of attacks on OPC- UA applications in the Industry 4.0 era [J]. Zhongguo Jixie Gongcheng/China Mechanical Engineering, 2019, 30(2): 220-225.
- [4] 秦仁凯. 基于 OPCUA 规范的数控系统数据统一接口的研究 [D]. 武汉:华中科技大学,2017.
- [5] 栾燕,张娟娟,张健. 工业网关分类及关键指标分析[J]. 自动化仪表,2020,41(7): 93-97.
- [6] 段宝燕. 智慧灯杆为载体的边缘智能网关研发 [D]. 重庆:西南大学,2020.
- [7] 魏福山,马传贵,程庆丰. 基于 RSA 的网关口令认证密钥交换协议[J]. 计算机学报,2011,34(1): 38-46.
- [8] 赵估. 边缘计算网关设备的构建和部署问题的研究 [D]. 大连:大连理工大学,2018.
- [9] 于旭冉. 基于 OPCUA 的工控信息化数据网关关键技术的研究与实现 [D]. 北京:北京邮电大学,2019.
- [10] 王静如. 基于边缘智能网关的家庭看护系统设计与实现 [D]. 武汉:华中科技大学,2019.
- [11] 雷振伍,达呼. 基于 HBase 与 OPC UA 技术的工业云架构研究 [J]. 工业控制计算机,2018,31(8): 95-96,99.