IoTDB 物联网数据库在城市轨道交通车辆智能运维系统中的应用

姜仕军[1]　徐晓晨[1]　徐燕芬[1]　杜广林[2]

(1. 中车青岛四方车辆研究所有限公司,266031,青岛;2. 青岛地铁集团有限公司运营分公司,266041,青岛 //第一作者,工程师)

摘　要　介绍了城市轨道交通车辆智能运维系统的功能需求,并分析了 IoTDB 物联网数据库时序数据库在构建城市轨道交通车辆智能运维系统中的适用性。基于 IoTDB,提出了一种轻量化城市轨道交通车辆智能运维系统总体架构,该架构能够有效提高系统对列车时序数据的处理能力和处理效率,降低系统运维难度和成本。

关键词　城市轨道交通; 车辆; 智能运维; 物联网数据库

中图分类号　U29-39;U279.2

DOI:10.16037/j.1007-869x.2021.09.045

Application of IoTDB in Intelligent Operation and Maintenance System for Urban Rail Transit Vehicles

JIANG Shijun, XU Xiaochen, XU Yanfen, DU Guanglin

Abstract　Functional requirements of intelligent operation and maintenance system for urban rail transit vehicles are introduced, and the applicability of a time-series database of IoTDB in the construction of the system is analyzed. Based on IoTDB, a lightweight overall architecture of intelligent operation and maintenance system for urban rail transit vehicles is proposed. This architecture can effectively improve the processing capacity and efficiency of timeseries data generated by trains, reducing the difficulty and cost of system operation and maintenance.

Key words　urban rail transit; vehicle; intelligent operation and maintenance; IoTDB

Author's address　CRRC Qingdao Sifang Rolling Stock Research Institute Co., Ltd., 266031, Qingdao, China

城市轨道交通(以下简称“城轨”)作为大中城市公共交通的主动脉,每天客流量高达几百万人次,且客流量还在不断上升,这对各城市的轨道交通公司的运维能力提出了较高的要求:一方面,要保障的线路安全可靠运行,避免发生安全事故;另一方面,要优化维修计划,将“计划修”转变为“状态修”,从而减少车辆维修时间,降低维修成本[1]。因此,需要采用一种智能化的城轨车辆运维方案,实现对城轨列车关键系统和部件运行状态的实时监测,并依托大数据、人工智能等技术,结合车辆运行和检修数据进行分析挖掘,诊断并预测设备的健康状态,从而保障的车辆安全性和可靠性。

在城轨车辆智能运维系统设计中,数据的采集、存储和管理是实现系统其它功能的基础。目前,城轨车辆智能运维系统大多以关系型或非关系型数据库作为其数据存储的核心架构。这种数据库虽然实现了时序数据的存储需求,但写入和查询性能较差,且在数据压缩、数据展示等方面功能不够完善。时序数据库是近年来在物联网领域内十分流行的一种数据持久化方案,其具备高性能的数据处理能力、高效的压缩算法和符合时序特征的存储引擎[2],广泛应用于对物联网设备实时数据的采集、存储和展示。

本文基于 IoTDB(物联网数据库)时序数据库的特点和优势,研究了其在城轨车辆智能运维系统构建中的适用性;并将其作为系统的核心部分,设计了一种轻量化的系统架构,有效提高了存储空间利用率和数据检索效率。

1　IoTDB 时序数据库

时序数据库的全称为时间序列数据库(Time-Series Database),主要用于处理带时间标签的数据,即时间序列数据。传统的关系型数据库、非关系型数据库由于天生的劣势,导致其无法满足对时序数据的高效存储与处理。相比之下,时序数据库在实时性、稳定性、兼容性和安全性等方面都具有更高的技术水平,又能和大数据生态进行融合,具有良好的应用前景[3]。

IoTDB 是清华大学自主研发的一款时间序列

数据库,实现了对时间序列数据收集、存储和分析的一体化管理,具有体量轻、性能高、易使用的特点,适用于工业物联网应用中海量时间序列数据高速写入和复杂分析查询的需求。目前,IoTDB 支持布尔型(Boolean)、32 位整型(Int32)、64 位整型(Int64)、单精度浮点型(Float)、双精度浮点型(Double)、字符串型(Text)等 6 种数据类型,具有良好的软硬件兼容性,支持多种硬件架构的部署,包括嵌入式终端、边缘计算设备以及数据中心服务器,并能与 Hadoop、Spark 生态完美对接,时序数据采集频率可达毫秒级。此外,IoTDB 的数据压缩能力是其相比于关系型数据库的一大优势,通过基于 Snappy 算法的无损压缩方式,能够有效节省存储介质空间,提升数据检索效率。

2 城轨车辆智能运维系统需求

目前,北京、上海、深圳等地的城轨公司已开展了针对轨道交通运维智能化与健康管理方面的相关工作[4]。城轨车辆智能运维系统需要实现从数据采集、数据存储到数据分析、数据展示的全流程、全功能的覆盖[5-6]。流转于系统内的数据具有变量多、周期短、变化小、时效性强等特点,因此对系统中各功能模块的性能提出了较高要求。

2.1 实时数据接收

城轨车辆上安装了数据采集和发送装置,实时采集和发送车辆的运行状态数据和故障数据。城轨车辆智能运维系统可实时获取列车信息。为支撑实时监控需求和数据分析需求,数据发送周期一般为毫秒级,系统需在一个周期内完成对所有车辆发送的当前周期内数据的校验和存储操作。

2.2 数据存储

一条城轨线路的列车数量一般在 20 列以上,每列列车上各子系统的传感器数量可达上万个,需回传至地面系统的变量一般为 3 000~5 000 个左右,每个变量至少需存储的内容包括标识符、时间戳和值。按照 500 ms 的发送周期、每个变量占 14 字节(标识符占 4 字节,时间戳占 6 字节,值占 4 字节)的存储空间来计算,城轨车辆智能运维系统覆盖一条线路时,每条线路按 20 列列车、3 000 个变量/列车,一年所需的存储空间为 52 980 480 000 000 字节,约为 48.19 TB。当覆盖的线路增多时,数据量将会呈线性增长。

2.3 数据查询

实现城轨列车运行状态的实时监控是城轨车辆智能运维系统的核心功能之一,这就要求系统必须在一个数据接收周期内将所有变量当前的最新值更新至显示前端,满足用户对实时数据的监控需求。另一方面,系统还应提供对历史数据的查询和展示功能,即将一段时间范围内的数据以美观、易理解的图表形式向用户展示出来。除了要保证查询结果的完整性和准确性外,还要在用户可接受的时间内返回结果。

2.4 计算统计

城轨车辆智能运维系统所接收的数据中,有些需要先进行计算、换算或者统计,然后再进行展示。例如:通过电流值和电压值计算功率值、换算载荷值的单位,统计牵引能耗在某小时、某天、某月内的最大值、最小值、平均值、累计值,等等。若采用传统关系型数据库进行存储,则需要编写复杂的 SQL 语句才能实现上述计算功能。

3 IoTDB 适用性分析

3.1 应用架构体系

IoTDB 由多个组件构成,涵盖数据收集、数据写入、数据存储、数据查询、数据分析、数据可视化等多个功能。

IoTDB 应用架构如图 1 所示。IoTDB 通过 JDBC(Java 数据库连接)驱动,广泛地支持多种异构数据源的接入,包括设备数据、系统状态数据、消息队列数据、应用数据以及其它数据库中的数据等。用户通过命令行客户端交互工具能够对数据库进行写入和查询操作,也可以通过 Grafana 监控工具以图形化方式查看数据变化趋势。TsFile 是一种专门为时间序列数据而设计的存储格式,支持高效的压缩和查询能力,是 IoTDB 的核心组成部分。对于写入 TsFile 文件中的数据,可以通过 TsFileSync 同步工具将文件同步至 HDFS(Hadoop 分布式文件系统)中,进而实现在 Hadoop 或 Spark 等开源平台上

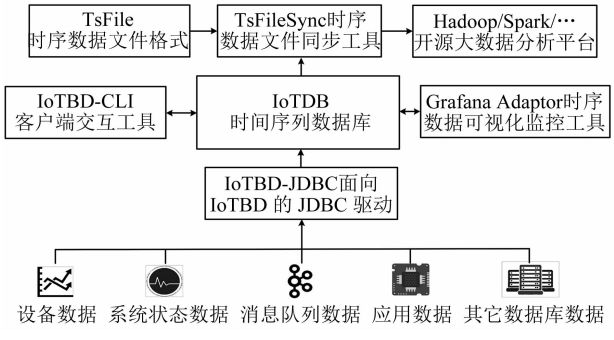

图 1 IoTDB 应用架构

进行时序数据的处理和分析。

3.2　性能及功能优势

在高速、海量时序数据存储方面，传统的关系型数据库难以达到低延迟、高吞吐量的实时存储，而且在查询大量历史数据时，会出现长时间无响应甚至引起系统崩溃等情况，严重影响系统稳定性和可靠性，降低了用户体验标准。IoTDB 作为新兴的时序数据库，对时序数据的处理具有天然的优势，能够实现每秒数百万数据点写入和查询的能力。

文献[7]将 IoTDB 与 HBase、MongoDB、Riak-TS、Redis 等数据库进行了对比测试，在 4 种实际应用场景下，分别对上述 5 种数据库进行同样的写入或查询操作，并统计了各自的执行时间以及工作负载。对比测试结果表明：针对大数据量的时序数据，IoTDB 具有明显的性能优势和稳定性。

本文通过模拟城轨车辆运行状态数据，对 IoTDB-v0.11.2 进行了写入和查询性能测试，测试结果汇总于表 1 和表 2 中。根据测试结果可以看出，IoTDB 能够有效支撑线网级城轨车辆智能运维系统的写入和查询性能需求。

表 1　IoTDB 写入性能测试结果

列车数/列	数据包点数/点	发送周期/ms	写入速度/(万点/s)
100	3 000	100	839.63
100	5 000	200	886.68
100	8 000	200	868.34
100	10 000	200	504.01

表 2　IoTDB 查询性能测试结果

并发数/个	列车数/列	数据点数/点	查询时间范围/h	耗时/s	查询速度/(万点/s)
1	1	20	24	1.09	1 585.32
1	5	20	24	5.42	1 594.10
10	5	20	12	4.81	8 981.29
10	5	20	24	6.04	14 304.64
50	5	5	24	4.86	22 222.22
50	5	10	24	6.07	35 584.84

注：查询测试是在有写入负载的情况下进行的。

此外，IoTDB 时序数据库还具有以下几个优点：

1）异常数据处理：由于网络延迟、软件性能、设备故障等原因不可避免地会出现数据无序到达、产生错误数据和重复数据等异常情况，IoTDB 能够支撑这些复杂的工业应用场景，包括时间序列数据的乱序写入、时间序列数据的批量更新，以及对无效、无用时间序列数据的清理删除。

2）数据降采样：是指数据库对查询到的结果集按照一定规则进行重新筛选，使筛选后的数据量小于原始数据量，且又不影响数据查询者的应用需求。例如，对于城轨车辆智能运维系统中根据设备运行数据绘制折线图的需求而言，如果返回的数据点数大于显示器分辨率的宽度，则不仅对于绘制图像的准确性没有任何帮助，反而还会造成数据点密级地堆叠在一起，影响整体显示效果。IoTDB 通过聚合操作实现数据的降采样功能，既能保证图表的准确性，也能有效减少数据传输量，提高响应速度。

3）灵活扩展：IoTDB 支持“一写多读”的部署模式，即一个系统内可以部署多套 IoTDB，其中，写入节点 IoTDB 负责新数据的写入和查询负载，其它多个查询节点的 IoTDB 只负责历史数据的查询负载。通过这种机制有效均衡了写入和查询工作量，避免两种操作对磁盘、网络的相互影响。随着数据量的不断增加，只需扩展查询节点的硬件设备，无需中断系统的正常运行。此外，最新版本的 IoTDB 基于 Raft 协议实现了一种分布式框架，将数据按时间序列组进行分区，以多副本的方式实现数据的可靠存储，并通过共识协议保证数据的强一致性[8]。

4）时序数据操作：城轨车辆上的子系统、设备、传感器种类繁多，各自具有不同的采样频率，在进行数据分析之前，需要对时序数据进行预处理。IoTDB 支持多种基于时间序列维度的数据操作，如按照时间戳进行数据对齐、按时间戳进行时序数据分割等，有效减少了数据预处理的难度和复杂度。

4　IoTDB 功能设计

4.1　存储结构

对实时数据和历史数据存储是时序数据库最基础、最核心的功能。在 IoTDB 时序数据库中，数据点是最小的数据单位，一个数据点由“时间戳-值(timestamp-value)”对组成。其中，“值”就是物理世界中传感器发送的数值。一个传感器所有数据点的集合，即为该传感器的时间序列数据。

对于待写入或待查询的数据点，需通过多层级的路径进行查找，例如，“root. a. b. c. d”即为 IoTDB 中的一条路径。其中，“root”为根节点，所有的路径都以“root”起始；“a”“b”“c”“d”为不同的层级的名称，层级 a 表示某个城市的一条城轨线路，层级 b 表

示该线路上的一列城轨列车,层级 c 表示该列车上的一个设备,层级 d 表示该设备上的一个传感器。图 2 展示了 IoTDB 的路径层级示例。通过“路径+时间范围”的组合,可以唯一确定 IoTDB 中的时序数据。此外,采用路径的层级设计,可以实现通过路径划分不同的存储空间,属于同一路径层级的数据能够存储在连续的磁盘空间上,避免了频繁的 I/O(输入/输出)切换,并且隔离了不同的时序数据。

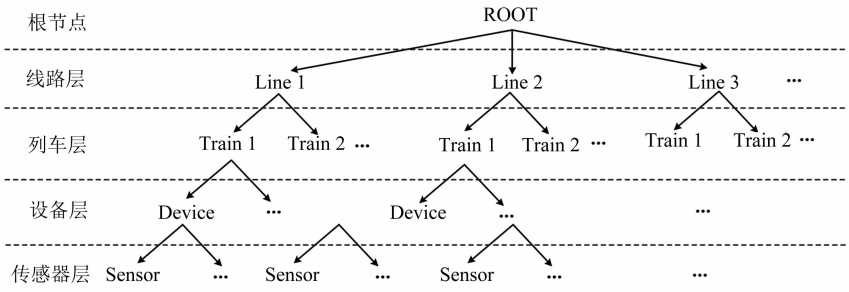

图 2 IoTDB 路径层级示例

4.2 数据压缩

城轨车辆智能运维系统需要占用庞大规模的数据存储空间,利用 IoTDB 的历史数据压缩能力可以有效减少历史数据的数据量,节省存储介质成本。所谓历史数据压缩,即利用各种算法缩小历史数据的冗余部分,同时尽量减少或避免数据失真。历史数据的压缩方式一般分为有损压缩、无损压缩和二级压缩三种[9-10]。其中,有损压缩能够实现较高的压缩比,但会导致数据精度下降;无损压缩不会降低原数据的精度,但要在压缩率、压缩速度和解压速度三者之间进行权衡;二级压缩则是结合了上述两种压缩方式的优点,即先对数据进行第一级有损压缩,再使用无损压缩算法进行第二级压缩。此外,压缩算法的效果还依赖于数据本身,数据变化越小、精度要求越低,则压缩效果越好。

4.3 数据计算

相比于关系型数据库,IoTDB 时序数据库能够提供更为强大的计算能力。通过 IoTDB 内置的统计分析计算函数,如求和、取平均值、计数等,可以根据时序数据的时间戳进行基于时间断面的计算、基于年月日的统计计算等。结合各类函数和自定义的计算公式,能够实现对原始数据进行复杂计算,计算结果可保存在 IoTDB 中,也可用于再次计算。

4.4 大数据分析

基于大数据技术和 Hadoop 生态软件进行城轨列车运维数据分析是当前的一个热门课题。IoTDB 能够完美对接 Hadoop 生态中的各种软件,配合 Hadoop 提供的分布式计算、存储机制,可提高城轨车辆智能运维系统在大数据管理和分析方面的运行效率和处理能力。此外,IoTDB 还可以对接 Spark 实时计算引擎,提供一种轻量级的数据分析解决方案,降低硬件资源部署量。

4.5 数据展示

存入 IoTDB 的时序数据需通过可视化工具进行展示,便于城轨车辆智能运维系统的用户对进入系统的原始数据进行观察和分析。Grafana 是一款开源的度量分析和可视化工具,具有数据监控、数据统计和告警等功能,常被用于展示设备运行的时序数据。通过开发 IoTDB-Grafana 适配器,用户可利用 Grafana 的 Web 页面以可视化图表的方式直接查看 IoTDB 中的数据,也可以在 Grafana 上进行一些数据探索工作。

5 城轨车辆智能运维系统架构设计

目前,各城市的轨道交通车辆在故障诊断和维修方面存在实时监测种类不全、过修欠修、故障误报等问题[11],信息化、智能化水平不高。因此,在保障城轨列车安全、可靠运行的基础上,尽量降低维修成本、提升城轨设备智能化管理水平,越来越成为轨道交通行业广泛关注和研究的热点[12]。

城轨车辆智能运维系统以保障城轨车辆运行安全、提高车辆检修质量、提升运营管理整体效能为目标,结合物联网、云计算、大数据等技术,实现对列车运行过程的全息感知和实时监控,有效辅助管理人员进行科学决策。基于 IoTDB 时序数据库构建城轨车辆智能运维系统,其总体架构如图 3 所示,共分为 3 层,包括数据源层、数据存储层和数据应用层。该设计以 IoTDB 时序数据库代替了传统的关系型数据库和 NoSQL 数据库,显著提高了对

城轨列车时序数据的写入和查询效率，且能够满足数据量持续增长的需求。

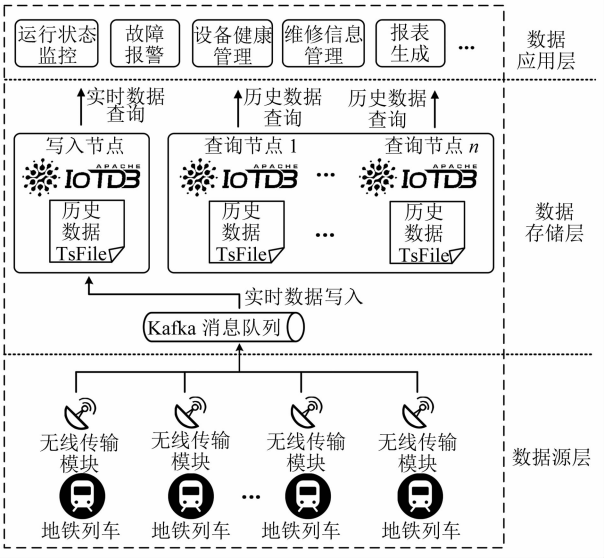

图 3　基于 IoTDB 的城轨车辆智能运维系统总体架构

数据源层覆盖所有城轨列车，列车上不同子系统、不同设备上的传感器是数据产生的源头，这些数据按照特定发送周期，通过无线传输模块以 TCP、MQTT 等协议发送至城轨运营公司的数据中心。

数据存储层实现了整个城轨车辆智能运维系统的数据汇集和持久化，主要由 IoTDB 时序数据库和 Kafka 消息队列组成。城轨列车发送来的时序数据首先进入 Kafka 消息队列进行缓存，按照一定的规则或算法进入不同的 Topic 和 Partition，这样既能分担写入任务的负载，也能通过 Kafka 的副本机制，确保接收到的数据不会丢失。IoTDB－JDBC 接口从 Kafka 的消费者端接收列车的实时数据，并存入写入节点的实时数据 TsFile 文件中。随着数据量的不断扩大，当单个 IoTDB 节点的存储能力无法支撑数据存储时，可采用横向扩展的方式再部署一个或多个 IoTDB 查询节点，并设置为只读模式。在“一写多读”方式下，为避免单点故障，实现高可用，将写入节点配置为主备模式，通过 IoTDB 自身的同步机制实现数据同步。实时数据为监视控制类应用提供支撑，历史数据为数据分析和挖掘类应用提供训练和测试样本。由于采用了数据压缩技术，历史数据所占用的存储空间能够得到有效控制。

应用层是系统对外输出能力的展现，能够提供如车辆运行状态监控、故障报警、设备健康管理、维修信息管理、报表生成等多种应用。城轨运营公司基于这些应用可以实现智能化的管理，减少人力成本，提高城市轨道交通服务水平。

6　结语

以 IoTDB 时序数据库为核心构建城轨车辆智能运维系统，可以充分发挥其处理城轨列车时序数据的天然优势，同时又可以无缝对接大数据管理分析平台，具有高性能、可靠性和易用性等特点。本文给出的轻量化系统架构设计，可为城轨车辆智能运维系统的开发提供参考和借鉴。

参考文献

[1] 刘述芳. 城市轨道交通关键设备智能运维系统初步建构[J]. 设备管理与维修，2018(增刊 1)：22.

[2] 徐化岩，初彦龙. 基于 influxDB 的工业时序数据库引擎设计[J]. 计算机应用与软件，2019(9)：33.

[3] 王妙琼，魏凯，姜春宇. 工业互联网中时序数据处理面临的新挑战[J]. 信息通信技术与政策，2019(5)：4.

[4] 侯文军，吴彩秀. 地铁车辆智慧运维平台研究[J]. 电力机车与城轨车辆，2019(6)：1.

[5] 张健，张钰薇，韩振. 轨道交通车辆智能化生产与运维系统的发展与探索[J]. 科技创新与应用，2019(27)：78.

[6] 张唯. 车辆智能运维建设需求与框架设计研究[J]. 现代城市轨道交通，2019(6)：10.

[7] LOU Y S，QIN Y，YE F，et al. Hydrological stream data pipeline framework based on IoTDB[J]. Service Oriented Computing and Applications，2019，13(4)：9.

[8] 李天安，黄向东，王建民，等. Apache IoTDB 的分布式框架设计[J]. 中国科学：信息科学，2020(5)：621.

[9] 戚雪冰. 风电监控系统中时序数据管理系统的设计与实现[D]. 南京：东南大学，2015.

[10] 辛晃，易兴辉，陈震宇. 基于 Hadoop+MPP 架构的电信运营商网络数据共享平台研究[J]. 电信科学，2014(4)：135.

[11] 詹炜，徐永能，王依兰. 城市轨道交通车辆智能运维系统应用研究[J]. 城市公共交通，2018(12)：28.

[12] 刘丙林，朱佳，李翔宇. 城市轨道交通车辆智能运维系统探索与研究[J]. 现代城市轨道交通，2019(6)：16.

（收稿日期：2020－09－07）