

新能源客车数据监管平台设计与实现

叶标^{1,2}, 冯拔^{1,2}, 李祥^{1,2}

(1. 长沙中车智驭新能源科技有限公司, 长沙 410083;

2. 中车时代电动汽车股份有限公司, 湖南 株洲 412007)

摘要:为解决企业新能源客车数据监管平台因海量终端接入带来的高并发压力和数据处理响应时间长的问题,本文开发一种新型监管平台,以CAN总线技术采集车辆数据,基于Netty架构搭建设备通信网关,使用Kafka中间件将业务解耦,通过SpringCloud微服务框架管理项目。结果表明,该新型平台能满足企业对新能源车辆数据的监管要求。

关键词:新能源客车; 数据; 监管平台

中图分类号:U495; TP311.1

文献标志码:A

文章编号:1006-3331(2023)01-0027-06

Design and Implementation of Data Supervision Platform for New Energy Buses

YE Biao^{1,2}, FENG Ba^{1,2}, LI Xiang^{1,2}

(1. Changsha CRRC Intelligent Control and New Energy Technology Co., Ltd., Changsha 410083, China;

2. CRRC Electric Vehicle Co., Ltd., Zhuzhou 412007, China)

Abstract:In order to solve the problems of high concurrency pressure and long response time of data processing caused by the access of massive terminals in data supervision platform for enterprise new energy buses, this paper develops a new supervision platform, which collects vehicle data with CAN bus technology, builds equipment communication gateway based on Netty architecture, uses Kafka middleware to decouple the business and manages projects by SpringCloud microservice framework. The results show that the new platform can meet the enterprises' supervision requirements to new energy vehicles data.

Key words:new energy bus; data; supervision platform

随着新能源车辆持续增涨^[1],为满足GB/T 32960.3—2016《电动汽车远程服务与管理系统技术规范 第3部分:通讯协议及数据格式》和国家及地方对新能源客车数据进行实时监管的要求,各车企纷纷搭建自己的车辆数据监控平台^[2]。随着平台接入车辆数据体量和种类的逐年增长,以及为实现对车辆关键零部件数据的实时、不间断采集,车辆数据发送频率、流量也在不断增加,使车辆监管平台面临高并发压力以及数据延时长等问题^[3]。基于此,本文设计和实现一个能解决上述问题,同时还能提高业务能力、兼具更优扩展性的车辆监控平台。

1 相关技术

本文平台通过CAN总线技术采集车辆数据,基于Netty网络应用程序架构^[4]搭建通信网关,利用Netty架构异步非阻塞^[5]、事件驱动等特性,解决大量车载终端连接的高并发压力,同时将耗时的数据处理工作绑定自定义线程池进行异步处理;复杂的数据处理业务交由SpringCloud微服务集群进行分布式处理,通过优化线程池参数、自定义解码和编码方式、自定义通信协议,进一步提升平台的高并发性和低时延性。

1.1 Netty 通信架构

Netty框架是基于Reactor模型^[6]开发的,通过异

收稿日期:2022-10-10。

第一作者:叶标(1994—),男,硕士;工程师;主要从事新能源车辆远程监控平台开发、车载终端数据分析工作。E-mail:404871394@qq.com。

步事件驱动和 epoll I/O 多路复写技术实现终端设备的高并发接入,其模型如图 1 所示。Netty 服务端推荐采用主从多线程池模型,主线程池 (BossGroup) 和从线程池 (WorkGroup) 都是 NioEventLoopGroup 类型,主要功能是维护 EventLoop 子线程和 Selector 选择器;EventLoop 线程通过选择器,轮询与其绑定的 Socket 的网络通信,并执行处理任务。其中主线程池用来处理设备的连接,将建立的 Socket 信道注册到 WorkGroup 线程池中 NioEventLoop 子线程中的 Selector 中,从而实现与 Worker 线程进行绑定。从线程池处理 Socket 信道数据读写,数据读写任务完成后,其他任务相继根据注册到 NioServerSocketChannel 的顺序依次进行。当存在多个设备与服务器建立连接时,服务器通过 Selector 监听每个信道上是否需要读写。这种非阻塞的读写过程不仅能提升 I/O 线程的运行效率,同时能够实现线程并发进行多个设备连接和数据读写操作。Netty 服务端架构^[7]包含 Boss 和 Worker 两个事件循环组 (NioEventLoopGroup),Boss 组主要负责处理连接事件并将请求分发给 Worker 组,由 Worker 组负责 IO 读写和业务处理。

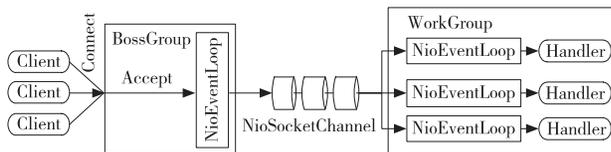


图 1 网关 Netty 模型图

1.2 Kafka 信息中间件模型

Kafka 作为信息中间件模型^[8]有两种信息传递方式:点对点的传递模式和分布式的发布-订阅模式。本文设计的监控平台包含多个数据处理业务,每个业务模块包含一个消费者,符合多个消费者消费一个 Topic 的应用场景,因此本文使用分布式的发布-订阅模式。

Kafka 作为一个分布式的发布/订阅信息传递的中间件模型,具有可扩展、高吞吐量、有序化、持久化等特性,常用于处理异步通信、应用解耦以及流量控制等问题。同时 Kafka 模型的分布式体现在 Kafka 采用集群的方式在多台 Kafka 服务器上运行,并通过

Zookeeper 对集群进行管理^[9],支持动态水平扩展,同一个 Topic 的消息可以存储在不同的 Partition 分区来提高吞吐率。在消费者读取信息时,每个分区为信息标记位置,并通过偏移量记录当前消费信息的位置,从而保证每个 Partition 内的信息有序。Kafka 从底层看是一条信息队列,并将存入队列的信息数据进行持久化,在 Kafka 处理系统明确指出数据已经被处理完之前,都可以将数据恢复,消费者通过请求 Broke 中转站,从 Topic 中获取数据。

1.3 CAN 总线技术

CAN 总线技术 (ControllerArea Network)^[10],作为广泛应用在客车内部控制器之间的通讯协议,具有通信及时性高、抗干扰强、传输误码率低、成本低等特点。多个控制模块通过 CAN 控制器连接到 CAN 总线网络,CAN 总线通过广播的方式,将其报文数据传输到各个 CAN 模块,每个报文具有 8 个字节长度和唯一标识符 ID。同时车载终端设备通过两根 CAN 总线连接到 CAN 总线网络并获取客车的电机、电池、电控三电数据以及其他相关零部件的控制信息。

1.4 SpringCloud 架构

基于 SpringCloud 的微服务架构^[11]将车辆监控平台的不同功能模块拆分成相互独立的微服务应用,每个微服务应用专注于单一的业务功能。微服务应用又是基于 SpringBoot 架构搭建,相互之间通过轻量级 Http 协议进行通信,易于开发和运维。

2 系统设计与构建

2.1 系统总体架构设计

本文设计的新能源客车数据监管平台,通过 CAN 总线技术采集整车关键零部件信息,基于 Netty 网络通信开发架构和自定义编解码方式,实现车载终端数据的采集与转发,实现终端到网关之间的可靠传输。引入 Kafka 信息中间件,基于其异步通信的特性提高网关的并发量;利用应用解耦的特性将数据流式处理模块与存储分析模块和业务模块进行解耦;基于 B/S 模式构建 Web 应用平台对车辆数据进行监测。平台总体架构如图 2 所示。新能源客车数据监管平台分为 4 个核心功能模块。

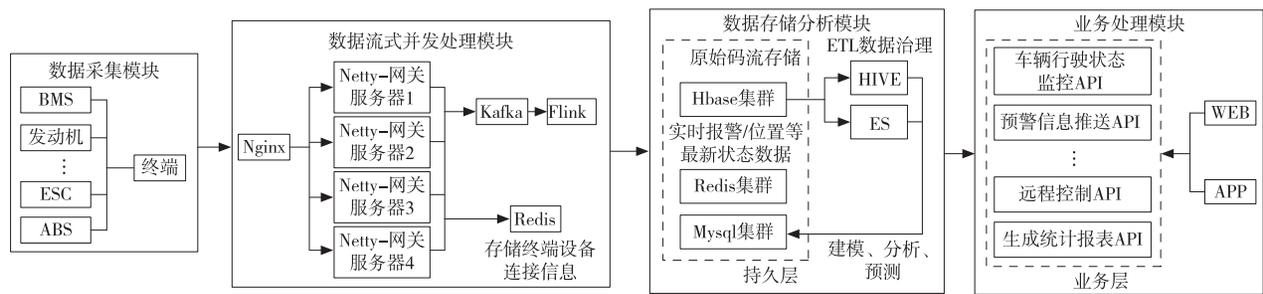


图 2 系统总体架构图

1) 数据采集模块。基于车载终端 T-Box, 通过 CAN 总线连接整车控制器, 采集车辆电机、电池、电控的核心三电信息数据, 是转发系统的数据来源。

2) 数据流式并发处理模块。基于 Netty 服务端模型搭建的网关, 通过 Redis 缓存终端设备与服务器之间的连接信息, 可支持扩展 GB 32960 以外的其他设备通讯协议。终端与网关建立连接 SocketChannel 的信道, 上传采集数据, 终端上传的数据为十六进制原始码流, 网关接收后存放在 ByteBuf 缓冲对象中。在传输过程中, 可能会发生 TCP 粘包的问题, 针对此问题, 通过自定义 Decoder 解码器^[9], 解析数据中标志本包数据单元长度的对应字节位, 将解析出来的数据长度与本包数据的实际长度进行对比, 如果长度不够, 则通过重置 ByteBuf 索引位置, 与下一包数据合并, 直到读取满足数据单元长度规定的完整的一包数据, 再将此包数据传递到下一个事务处理器 (Handler) 中。当有完整数据进入 Handler 时, 首先对数据内容进行解析, 解析出终端对应的车架号、数据采集时间等字段组成关键字, 然后通过调用 Kafka 的生产者 (Producer) 将数据存入 Kafka 中, 再通过 Flink 流式计算技术对原始报文数据进行实时解析处理, 并根据解析后数据特性进行分类, 存入不同的数据集。

3) 数据存储分析模块。包含 Hbase、Mysql、Redis 数据库, 分别以集群的方式部署并通过 Zookeeper 进行监管, 其中 Hbase 存储 Flink 计算后的数据集以及未处理的原始报文, 采用 HIVE、Hbase、Imapla 结合的方式, 配置 HIVE 与 Hbase 数据同步, 通过 Kettle 工具进行数据清洗, 并基于 Elasticsearch 进行建模分析。Redis 集群存储实时性要求较高的报文信息, 如实时报警信息、车辆定位信息等。Mysql 集群用于存储大数据分析过后生成的预警信息和运营统计报表

信息。

4) 业务处理模块。基于 SpringCloud 架构开发, 单个应用服务又是基于 Springboot 架构, 通过集成大量的框架, 并尽可能地自动配置 Spring, 以简化整个项目的搭建和开发过程。各服务之间使用 RESTFUL API 相互协调, 通过 Eureka 注册中心获取和管理服务, 通过 Ribbon 维持各个分布式服务器间的负载均衡, 通过 Hystrix 熔断机制保护微服务链路。前端使用 Vue 架构, 基于 MVVM^[12] (Model View ViewModel) 的模式, 实现后端传递数据与前端页面的双向绑定。将项目使用到的前端静态资源如 js、css 等单独存放在静态资源服务器上由前端维护, 从而实现前后端分离。业务处理模块的主要功能有:

①车辆行驶状态监控。车辆通过终端与平台建立通信连接后, 平台用户可以对车辆实时行驶状态数据进行监控。通过 WebSocket 客户端处理上传数据并发送给 WebSocket 服务器, 再通过 WebSocket 服务器实时推送给平台。

②预警信息推送。平台通过大数据建模对车辆关键数据进行分析、判断, 如车辆上传数据达到预警阈值, 则按照特定的规则生成预警信息, 并通过语音、短信、APP 的方式推送给车辆相关负责人。

③远程控制。车辆通过终端与平台建立通信连接后, 平台用户可以对车辆下发控制指令, 通过获取终端设备与 Netty 网关服务器间的连接信息, 然后根据控制信息内容封装成原始码流, 再通过 Netty 网关下发给车载终端, 从而实现车辆远程控制。

④生成统计报表。根据大数据模型生成车辆运营里程统计报表、故障统计报表、累计能耗统计等信息, 平台通过设置统计时间纬度生成统计报表, 同时支持报表导入导出。

2.2 通信架构设计

基于 Netty 服务端框架开发与设备进行通信的网关,通过异步事件驱动实现终端设备的高并发接入。终端与网关建立连接后,开始接收终端上传的数据,数据通过服务端非阻塞 I/O 模型的 Socket 通道进行读写,数据读写任务完成后,其他任务根据注册在 SocketChannel 的顺序依次进行。这种非阻塞的读写过程不仅能提升 I/O 线程的运行效率,同时能实现线

程并发进行多个连接的和数据读写操作。

Netty 的服务端架构在处理设备连接以及 I/O 读写处理后会触发 ChannelPipeline 中的自定义 Handler 线程池,通过异步的方式进行终端设备信息鉴权、消息回复等业务处理,同时根据消息类型将数据存入不同的 Topic 中,由后续微服务业务线程消费数据进行业务流程,数据处理流程如图 3 所示。

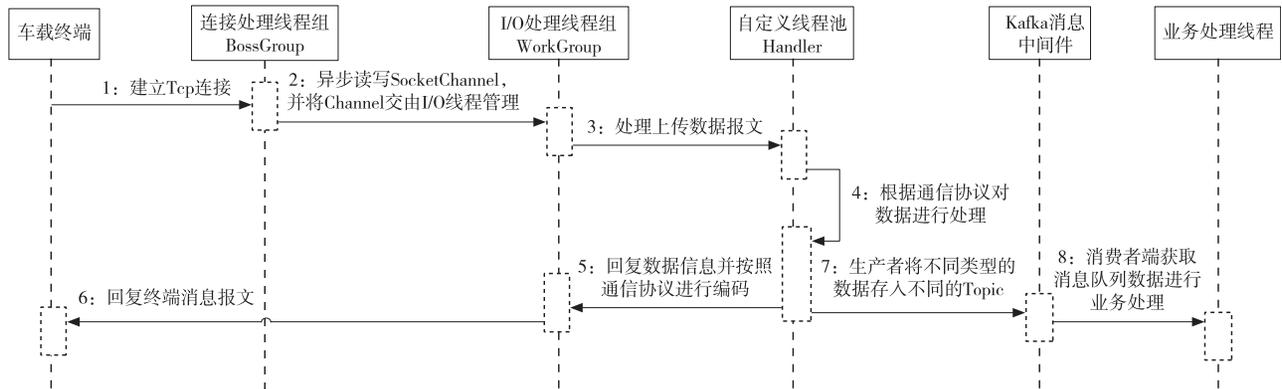


图 3 数据处理流程图

2.3 数据通信协议设计

在本文设计的平台中,终端与 Netty 服务端之间以国标 GB/T 32960 协议作为通信协议传输国标数据,协议格式见表 1;并基于此协议扩展了企标数据通讯传输协议,协议格式见表 2。表 2 中的企标数据包数据结构

封装在表 1 国标协议数据包中的数据单元部分。两种协议都是以 TCP/IP 网络控制协议作为底层通信承载协议。数据接收端在接收到传输数据后应先对数据格式进行判断,如果符合协议格式要求,则进行数据业务处理;如果不符合则丢弃数据,并断开连接。

表 1 国标协议数据包结构

字节	定义	数据类型	描述及要求
0	起始符	STRING	固定为 ASCII 字符‘##’,用“0x23,0x23”表示
2	命令单元	命令标识	BYTE
3		应答标志	BYTE
4	唯一识别码	STRING	车辆 VIN,符合 GB 16735
21	数据单元加密方式	BYTE	0x01:数据不加密;0x02:数据经过 RSA 算法加密;0x03:数据经过 AES128 位算法加密;其他预留,建议以 AES128 位加密
22	数据单元长度	WORD	数据单元的总字节数,有效值范围:0~65531
24	数据单元	-	
末字节	校验码	BYTE	采用 BBC(异或校验)法,校验范围从命令单开始,到校验码前一位

表 2 企标数据包数据结构

企标数据包数据结构组成	描述
Message_Header	数据包包头,固定长度
TBOX_Message_Body	终端数据包包体,变长,可为空
CAN_Message_Body	CAN 网络数据包包体,变长,可为空
Message_Ending	数据包包尾(校验),固定长度

2.4 性能优化

为进一步提升转发系统的性能,本文通过以下措施进行优化:

- 1) 将数据解析、持久化等耗时操作,加入异步线程池。

2) 通过自定义 Handler 并将其分配到独立的线程池, 然后将线程池以及 Handler 添加到 ChannelPipeline 容器中, 实现 Handler 的自定义线程池与 WorkGroup 线程池完全分离, 从而最大限度避免长时间程序阻塞。同时自定义线程池的大小, 设置最大线程数 N 的参考算法^[13]如下:

$$N = N_{cpu} \times U_{cpu} \times (1 + W/C)$$

式中: N_{cpu} 为核心 CPU 数目, 通过 `Runtime.getRuntime().availableProcessors()` 获得; U_{cpu} 为 CPU 使用率, 一般设为 1; W 为线程阻塞时间; C 为线程计算时间。

本文自定义线程属于计算密集型线程, 结合实际运行环境, 设置 N 为 $N_{cpu} + 1$ 。

3) 调整 Netty 服务端与客户端的参数设置: 设置 `TCP_NODELAY` 来关闭 nagle 算法, 避免数据包堆积, 从而降低延迟; 设置 `SO_RCVBUF`、`SO_SNDBUF` 来调整接收方和发送方的收发缓存区大小, 避免拥堵。

3 测试与结果分析

3.1 网关通信性能测试

使用 Jmeter 模拟车载终端设备以 10 s 一包的频率向网关发送报文。通过设置并发线程数为 3 000, 轮发请求间隔为 10 s, 重复 30 次, 然后在网关程序中定义一个计数器, 统计所有连接的通道数。测试结果如图 4 所示: 当并发量为 3 000 时, 通过对 30 次试验取平均值, 本文采用的自定义编码解码方式网关与采用传统 Java 序列化的网关每秒最大接入量分别为 2 047 台和 1 868 台, 相较之下基于自定义编码的网关接入量提升了 9% 左右。

[2021-08-08 11:09:18 366] 报告
 [2021-08-08 11:09:18 367] 信息: 收到平台登入报文, 身份信息正确!
 [2021-08-08 11:09:18 486] IP: 127. 0. 0. 1
 [2021-08-08 11:09:18 486] Port: 8 889
 [2021-08-08 11:09:18 486] 报告完毕
 [2021-08-08 11:09:18 487] 当前连接数: 1 287

图 4 网关并发量统计图

自定义 Handler 时将解析数据写入 Kafka 的 Topic 中, 统计不同并发量下 300 s 内存入 Topic 中的数据量作为事物处理总数 (简称 TC), 将 TC 作为通信

性能测试的结果, 其统计数如图 5 所示, 处理请求效率平均提升 30% 左右。

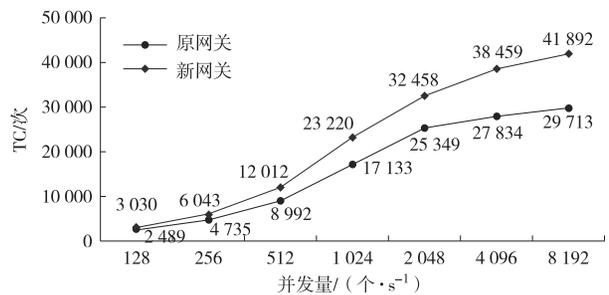


图 5 通信性能统计图

3.2 Web 应用平台响应测试

使用 Postman 设置 Monitor 监控任务对 Web 应用平台接口的响应时间, 并以视图的形式展示接口响应时间的测试结果。对 Web 应用平台接口进行单次 6 h 的监控, 共计 20 次, 监控平均响应时间为 0.74 s, 传统平台接口平均响应时间为 0.8 s, 相较之下, 平均响应速度提高了 8% 左右, 部分 Web 平台响应测试结果如图 6 所示。

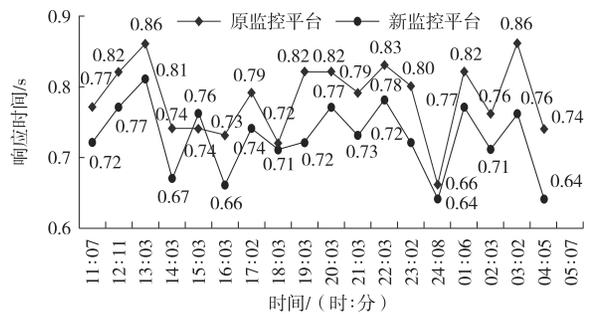


图 6 Web 应用平台响应测试统计图

4 结束语

本文提出了一种新能源客车数据监管平台设计方法, 解决了监控平台在实际生产环境中面临的高并发压力和数据延时长的问题, 提升了平台业务处理能力和接口请求响应速率, 适用于包含 GB/T 32960 和企标的多种协议, 满足企业日常监管新能源客车上传数据业务需求的同时, 具有较好的扩展性。

参考文献:

[1] 何亮. 我国新能源汽车保有量已突破 1000 万辆 [N/OL]. 科技日报, (2022-07-07) [2022-10-10]. <https://m.gmw.>

- cn/baijia/2022-07/07/35866450.html.
- [2] 胡艳峰. GB/T 32960 标准分析及应用[J]. 汽车电器, 2018(9):49-52.
- [3] LIU Xiongfei, LIU Jiakang, LIAO Beiping, et al. Design of IoT Web Server Communication Platform based on Netty and WebSocket[C]//Proceedings of the 3rd International conference on Mechatronics Engineering and Information Technology(ICMEIT2019), 2019:165-171.
- [4] HUANG Yingzhen, LIU Zheyuan, QIAN Rangxin, et al. A File Distribution Mechanism based on Netty Communication Framework[J]. IOP Conference Series Materials Science and Engineering, 2021, 1043(3):032047.
- [5] 段楠. 异步非阻塞网络通讯技术研究[J]. 现代计算机, 2019(17):79-82.
- [6] 赵思远. Java NIO 与 IO 性能对比分析[J]. 软件导刊, 2021, 20(4):214-219.
- [7] 甄凯成, 黄河, 宋良图. 基于 Netty 和 Kafka 的物联网数据接入系统[J]. 计算机工程与应用, 2020, 56(5):135-140.
- [8] 高宗宝, 刘丽美, 张家铭, 等. Spark 平台中 Kafka 偏移量的读取管理与设计[J]. 软件, 2019, 40(7):118-122.
- [9] 黄毅斐. 基于 ZooKeeper 的分布式同步框架设计与实现[D]. 杭州:浙江大学, 2012.
- [10] 季丽华. CAN 总线技术在汽车电控系统中的应用及检测分析[J]. 电子世界, 2021(18):178-179.
- [11] 黄伟, 李振, 王梨, 等. 基于“云平台+微服务”架构的智能客服平台构建[J]. 信息技术与标准化, 2022(9):84-87.
- [12] 崔慧娟. MVVM 模式在 Android 项目中的应用[J]. 信息与电脑, 2021, 33(6):1-3.
- [13] 张娜, 史佳炳, 吴彪, 等. 基于 Netty 和 Kafka 的 IOT 终端服务系统设计方案[J]. 浙江理工大学学报(自然科学版), 2020, 43(2):240-245.

◆ 广告目次

厦门金龙旅行车有限公司	(封面)	奇瑞万达贵州客车股份有限公司	(中插一)
福建中科云杉信息技术有限公司	(封二)	成都客车股份有限公司	(中插二)
《客车技术与研究》理事会名录	(封三)	北京福田欧辉新能源汽车有限公司	(中插三)
期刊信息	(封底)	太原泰立机电新技术有限公司	(中插四)
中车时代电动汽车股份有限公司	(前插一)	比亚迪汽车工业有限公司	(后插一)
厦门金龙联合汽车工业有限公司	(前插二)	中通客车股份有限公司	(后插二)