

基于 gRPC 分布式数据通信的 地铁线网指挥中心平台设计与实现

钟建峰¹, 孟宇坤², 王石生², 韩 天¹, 蔡宇晶², 宣秀彬¹

(1. 北京经纬信息技术有限公司, 北京 100081;

2. 中国铁道科学研究院集团有限公司 电子计算技术研究所, 北京 100081)

摘 要: 地铁线网指挥中心平台以综合监视系统为核心, 以数据应用系统、运营执行系统为协同管理。由于关联的设备系统多、业务复杂, 海量数据交互的实时性、并发性、准确性要求高, 线网指挥中心平台的数据通信问题一直是一个难题。为解决该问题, 基于 Protocol Buffer 通信协议, 设计和实现了基于 gRPC 分布式数据通信的地铁线网指挥中心平台系统架构。远程过程调用 gRPC 机制是分布式系统和计算机网络中通信的新型机制, 在网络化运营模式下通过采集大量的设备数据、业务系统数据, 可实现平台数据的规范化处理和业务系统间的高内聚、松耦合通信, 提高城市轨道交通运营安全保障能力。

关键词: 城市轨道交通; gRPC; RPC; 线网指挥中心平台; Protocol Buffer; 分布式

中图分类号: U231

文献标志码: A

文章编号: 1672-6073(2023)02-0190-08

Design and Implementation of a Metro Network Control Center Based on gRPC Distributed Data Communications

ZHONG Jianfeng¹, MENG Yukun², WANG Shisheng², HAN Tian¹, CAI Yujing², XUAN Xiubin¹

(1. Beijing Jingwei Information Technologies Co., Ltd., Beijing 100081;

2. Institute of Computing Technologies, China Academy of Railway Sciences Corporation Limited, Beijing 100081)

Abstract: The proposed metro network control center is based on an integrated monitoring system at its core, and uses a data application system and operation execution system for collaborative management. Owing to a large number of associated equipment systems, complex businesses, and high requirements regarding real-time, concurrency, and accurate massive data interaction, the data communication of metro network control centers has always been a problem. To address this, based on the protocol buffer communication framework, a metro network control center architecture based on gRPC distributed data communication is designed and implemented. The remote procedure, called the gRPC mechanism, is a novel communication mechanism in distributed systems and computer networks that collects a large amount of equipment data, business system data, and standardized processing of platform data. Accordingly, it provides high cohesion and loose coupling communication between business systems and improves the operational safety guarantee characteristics of urban rail transit.

Keywords: urban rail transit; gRPC; RPC; network traffic control center; protocol buffer; distributed service

收稿日期: 2022-02-21 修回日期: 2022-05-27

第一作者: 钟建峰, 男, 硕士研究生, 工程师, 从事城市轨道交通信息管理工程研究, 99dxs@163.com

通信作者: 王石生, 男, 本科, 副研究员, 研究方向为城市轨道交通信息工程, wangshisheng@sina.com

基金项目: 中国铁道科学研究院集团有限公司科技研究开发计划项目(2051GCJ201)

引用格式: 钟建峰, 孟宇坤, 王石生, 等. 基于 gRPC 分布式数据通信的地铁线网指挥中心平台设计与实现[J]. 都市轨道交通, 2023, 36(2): 190-197.

ZHONG Jianfeng, MENG Yukun, WANG Shisheng, et al. Design and implementation of a metro network control center based on gRPC distributed data communications[J]. Urban rapid rail transit, 2023, 36(2): 190-197.

1 研究背景

近年来,城市轨道交通行业发展迅猛,地铁已经是一些大中型城市不可缺少的交通工具,这对城市轨道交通的安全、高效提出了更高的要求。地铁线网指挥中心平台应运而生,它是一个综合性的监管平台,能够根据线网的实时状态,实现各线路统一管理、协调运作,其基础是对获得的各专业子系统的运营数据、生产数据、安全管理数据进行有效的整合与挖掘^[1]。然而,线网指挥中心平台由于以综合监视系统为核心,集成了众多复杂的设备数据,同时又集成了数据应用系统、运营执行系统等众多业务管理系统数据,存在多系统数据通信缺乏联动、协调难度大、信息交换效率低下、可扩展性低等问题,严重制约了城市轨道交通系统的整体效能发挥^[2]。gRPC 提出了支持分布式系统与分布式实体间相互协作的新型通信合作机制,可以改善地铁线网指挥中心平台的实时性、并发性、准确性,以及提供良好的可扩展性。同时,gRPC 也是轨道交通一种新的数据通信模式,可用于形成基于大数据量通信的车站-线路-线网多层次决策体系^[3]。

2 数据通信方案现状分析

2.1 常用数据交互方式

地铁线网指挥中心平台数据分为结构化数据和非结构化数据(文档、图片、音频、视频等)。根据数据的时效性可以分为实时数据和非实时数据。

目前对数据进行通信交互的方式,常用的有 4 种:数据库方式、Socket 方式、消息中间件方式和 Web API 方式。

1) 数据库方式:涉及的关键技术是 ODBC、JDBC、ADO.NET、PDO。该通信方式可以统一使用 SQL 语言进行数据读取和存储。但是对于实时数据需要定时高频轮询数据库,会对数据库造成较大的访问压力,不适合高频读写的实时数据。

2) Socket 方式:涉及的关键技术是 TCP/IP、UDP。其优点是可以建立高效的长链接数据通信方式,适合实时数据通信。但是由于 Socket 通信采用字节流进行传输,并未对数据格式进行规范化处理,因此需要对传输数据进行解析,并且需要专门对数据流的拆包、粘包、丢包等情况进行处理,增加了程序的复杂度。

3) 消息中间件方式:涉及的常用组件是 Kafka、ActiveMQ、RabbitMQ、Mqtt。优点是可以形成消息队

列,数据不易丢失,扩展性好。缺点是大数据量的消息队列,容易出现消息阻塞,对非结构化数据处理不够友好。

4) Web API 方式:涉及的常用通信文件是 Json、XML。优点是调用灵活方便,适应 B/S 和 C/S 架构程序方式。缺点是对于实时数据的传输,也需要定时轮询调用,对 Web API 接口调用压力比较大。

在以往的地铁线网指挥中心平台,常用以上 4 种方式进行多种组合,形成平台复杂的数据交互,客户端、服务端需要开发多种与之对应的数据交互方式,造成数据没有规范统一,存在多系统数据通信缺乏联动、协调难度大、信息交换效率低下、可扩展性低等问题。亟需一个分布式 RPC 框架方案对多种数据交互方式进行统一优化管理。

2.2 几种典型的分布式 RPC 通信框架对比

目前典型的分布式 RPC 通信框架主要有 Motan、Dubbox、Thrift、gRPC、Rpcx。

1) 地铁线网指挥中心平台集成众多系统和数据通信方式,其特点决定需要选择与语言无关的 RPC 框架。由于 Motan、Dubbox 只支持 Java 语言,Rpcx 只支持 Go 语言,首先排除使用单一语言的 Motan、Dubbox、Rpcx 框架方案。

2) Thrift 和 gRPC 是跨语言的 RPC 框架,Thrift 与 gRPC 框架对比如表 1。由于地铁线网指挥中心平台的综合监视系统需要对行车、客流、电力、ISCS

表 1 Thrift 与 gRPC 框架对比

Table 1 Comparison between Thrift and gRPC frameworks

指标	Thrift	gRPC
开发语言	跨语言	跨语言
分布式服务治理	可以配合 zookeeper, Eureka 等实现	可以配合 etcd(go), zookeeper, consul 等实现
底层协议	TCP/HTTP/FRAME	HTTP2
消息序列化	thrift	protobuf
流式传输	不支持	支持,双向流式通信
负载均衡	Haproxy, zookeeper+客户端负载均衡等方案	负载均衡软件 HaProxy, Nginx 1.13.10+等
容错	Failover	具有 Failover 失效切换的容错策略
注册中心	zookeeper	Etcd, zookeeper, consul
性能	比 gRPC 快 2~5 倍	比 dubbox, motan 快
设计代码质量	生成代码太多	生成代码少、干净
语言之间的特性	很多,不同的语言间有差异	少,实现的很好
文档	非常少,而且很多特性没有说明	非常详细

(综合监控系统, 包含 PSCADA、AFC、FAS、BAS、PSD、PIS 等设施设备系统)等设备状态进行实时数据展示的平台, 对数据实时性要求高。从表 1 分析可知, 虽然 Thrift 性能比 gRPC 快, 但是 gRPC 支持流式数据传输, 并可以实现双向流式通信, 能够解决数据实时性要求高的问题, 而 Thrift 不支持流式传输。

3) gRPC 还拥有语言之间的实现差异少, 生成代码干净, 文档非常详细等优点。

综上所述, gRPC 框架是目前适合地铁线网指挥中心平台的最优 RPC 方案。

3 gRPC 概述

随着信息技术的飞速发展, 整个地铁线网指挥中心平台的数据量呈现井喷式增长。过去集中式部署应用的经验不再适用, 逐渐演化出了分布式架构, 又慢慢地演变成微服务、集群的架构, 在这个过程中产生了如注册中心等组件^[4]。借助这些组件, 应用才能在集群环境下实现服务发现、服务治理、负载均衡等基础功能, 继而在此基础上采用分布式 gRPC 远程过程调

用进行数据通信。

3.1 gRPC 相关概念及调用过程

gRPC 是一个现代的可以在任何环境中运行的开源高性能远程过程调用(remote procedure call, RPC)框架。它可以有效地连接数据中心内和跨数据中心的服务器, 并提供可插拔的支持, 以实现负载均衡、跟踪、运行状况检查和身份验证^[5]。它也适用于分布式计算, 可将设备、应用程序和浏览器连接到后端服务。gRPC 一开始由 google 开发, 是一款语言中立、平台中立、开源的 RPC 系统。

在 gRPC 里, 客户端应用可以像调用本地对象一样直接调用另一台不同机器上服务端应用的方法, 使得创建分布式应用和服务能够更加容易。与许多 RPC 系统类似, gRPC 也是基于以下理念: 定义一个服务, 指定其能够被远程调用的方法(包含参数和返回类型)。在服务端实现这个接口, 并运行一个 gRPC 服务器来处理客户端调用。在客户端拥有一个能够像服务端一样的方法存根。gRPC 调用过程如图 1 所示。

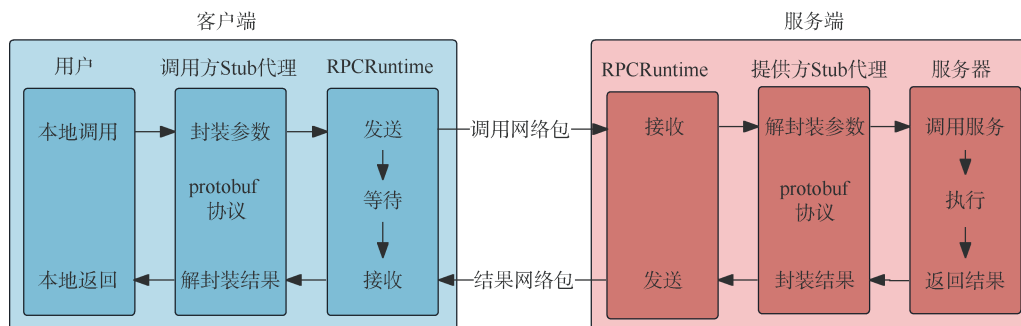


图 1 gRPC 调用过程

Figure 1 gRPC call procedure

3.2 gRPC 特点

1) 语言中立, 支持多种语言(C++、Java、C#、Go、Python、Ruby、Node.js、Android Java、Objective-C、PHP 等)。

2) 基于 IDL 文件定义服务, 通过 proto3 工具生成指定语言的数据结构、服务端接口以及客户端 Stub。

3) 通信协议基于标准的 HTTP/2 设计, 支持双向流、消息头压缩、单 TCP 的多路复用、服务端推送等特性, 这些特性使得 gRPC 在移动端设备上更加省电和节省网络流量。

4) 序列化支持 PB(Protocol Buffer)和 JSON。gRPC 默认使用 PB, PB 是一种与语言无关的开源的高性能

序列化框架, 用 proto 创建 gRPC 服务, 用 Protocol Buffer 消息类型来定义方法参数和返回类型, 基于 HTTP/2+ PB, 保障了 RPC 调用的高性能。

3.3 gRPC 适合的方案

gRPC 适合以下方案系统应用:

1) 微服务: gRPC 设计了用于低延迟和高吞吐量的通信, 对于效率至上的轻量级微服务非常有用。

2) 点对点实时通信: gRPC 支持双向流式传输。gRPC 服务可以实时推送消息, 无需轮询。适合需要流式处理请求或者响应的点对点实时服务。

3) 多语言环境: gRPC 支持大部分的常用开发语言, 适合使用多种语言开发的系统, 因此 gRPC 是多

语言环境的优先选择。

4) 简单的服务定义: 使用 Protocol Buffer 定义服务和数据通信格式, 这是一个功能强大的二进制序列化工具集和语言, 这使得 gRPC 在各个平台和实现中都是一致的。

5) 网络受限环境: gRPC 消息使用 Protocol Buffer 进行序列化, 因此 gRPC 消息始终小于等效的 JSON 消息和 XML 消息。

6) 快速启动并扩展: 使用单行安装运行时和开发环境, 并可以让使用框架每秒扩展到数百万个 RPC。

4 地铁线网指挥中心平台总体架构

地铁线网指挥中心平台集成了地铁运营、生产、管理过程的相关信息, 并通过可视化的综合监视系统进行采集、发布和展示, 实现了对地铁运营相关的移动设备和固定设施的数据应用分析、应急事件处理过程的监控和应急救援指挥的决策支持, 如图 2 所示。目前线网指挥中心平台已成为地铁线网运营管理的核心平台, 实现了利用一流的信息技术手段来保障一流的地铁运营的目的^[6]。

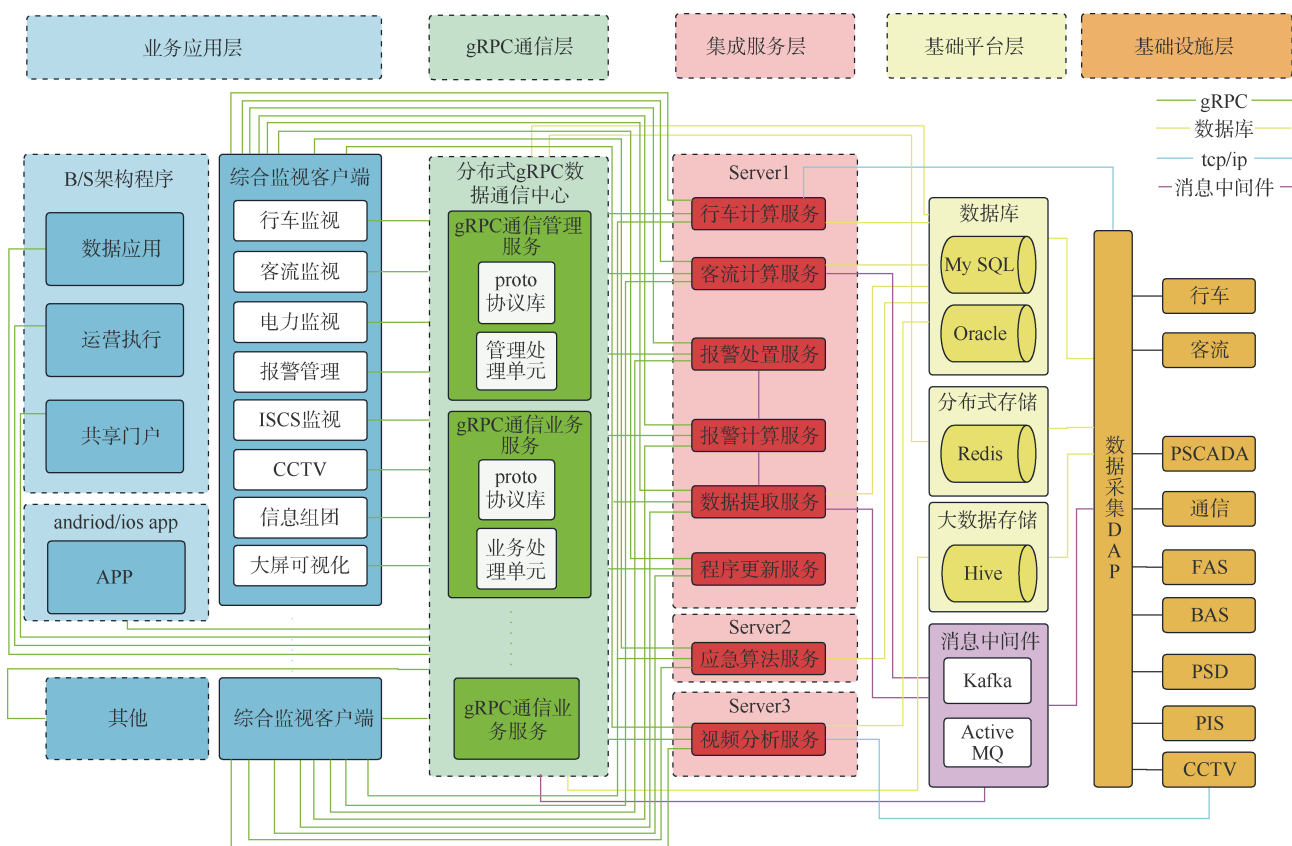


图 2 基于分布式 gRPC 数据通信的地铁线网指挥中心平台通信架构

Figure 2 Communications architecture of metro network control center based on the gRPC distributed data communications

1) 基础设施层。其为业务系统提供规范的信息基础设施服务, 包括网络设施设备、信息安全防护设施设备、存储及灾备设施设备等。基础设施层依托现有的综合监控系统、业务系统及应用集成平台, 通过数据采集中心 DAP 平台采集 ATS、客流、PSCADA、通信、FAS、BAS、PSD、PIS、CCTV 等系统设施数据, 形成以行车、客流、电力为指挥核心的综合运营指挥平台^[7]。

2) 基础平台层。依托数据库、分布式存储、大数据平台存储以及消息中间件平台, 实现多种数据源的

获取以及各专业数据的存储, 将轨道交通专业的数据资产管理与应用场景关联起来, 实现全面的应用与共享。

3) 集成服务层。集成服务层通过基础平台汇集大量有价值的信息, 依托微服务架构, 为各业务系统提供基础数据^[8]。实现指挥中心平台行车计算、客流计算、报警计算、程序更新、应急辅助算法、视频分析等服务体系构建。

4) gRPC 通信层。采用分布式 gRPC 通信技术, 可以跨平台、多语言之间进行通信, 构建主数据管理和

业务模型，形成管理处理单元和业务处理单元。gRPC 通信技术解决了信息多源异构，及大数据量、高并发量实时数据的传输效率和准确性问题，统一各应用系统和服务之间的通信协议。此外，利用分布式架构体系，实现数据标准转换、数据分享、数据管理、数据的多维度、多形式分析和应用。

5) 业务应用层。以综合监视系统为核心为城轨各个运营单元提供各类智能化的业务管理、运营管理和安全保障管理^[9]。在统一的操作平台上集成了各条线路的行车、客流、电力、报警管理、ISCS、CCTV 等监视模块，数据应用与数据分析模块，集中指挥与应急指挥模块，并实现了信息集中、协调指挥、应急发布、应急救援、综合统计、应用分析以及辅助决策服务^[10]。

4.1 分布式 gRPC 数据通信中心

分布式 gRPC 数据通信中心由 gRPC 通信管理服务和 gRPC 通信业务服务组成。图 3 是分布式 gRPC 数据通信中心架构示意。

1) gRPC 通信管理服务：主要负责 gRPC 自身运行需要的一些配置文件、缓存数据、基础数据。管理服务包含：系统单元注册信息、心跳保活信息、服务端信息、客户端连接信息、行车服务信息、客流服务信息、报警队列信息。

2) gRPC 通信业务服务：主要负责 gRPC 与各设备子系统的业务交互通信，另外与数据应用分析系统、

运营执行系统数据共享和数据联动，业务服务包含：基础信息业务通信服务(地铁线路信息数据、地铁车站信息数据、地铁区间信息数据、地铁列车信息数据、地铁用户信息数据、地铁系统菜单数据)、行车信息业务通信服务、客流信息业务通信服务(地铁网络三色状态运营信息系统客流业务、ACC 客流业务)、电力信息业务通信服务、告警信息业务通信服务、ISCS 信息业务通信服务、CCTV 信息业务通信服务、信息组团业务通信服务、大屏可视化业务通信服务。表 2 是几种典型 gRPC 通信业务服务实现的功能。

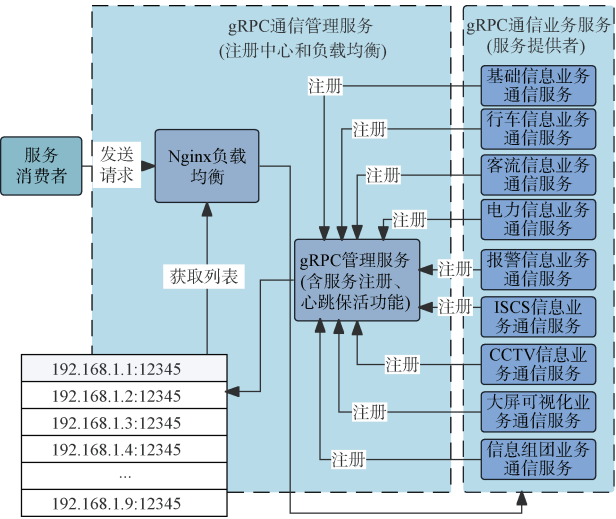


图 3 分布式 gRPC 数据通信中心架构示意
Figure 3 Architecture of the distributed gRPC data communication center

表 2 典型 gRPC 通信业务服务
Table 2 Typical gRPC communications services

gRPC 服务名称	gRPC 服务实现的功能
基础信息业务通信服务	线路基础信息，线路动态信息，线路主题信息，线路客流信息，线路应急事件处置信息，区间 TOS 信息，区间基础信息，车站基础信息，车站动态信息，车站客流动态信息，区间客流动态信息，车站出入口，车站设备，车站限流计划，四长联动信息，大客流响应信息，车站关站计划，限流车站统计信息，车站客流排名，列车基础信息，列车行驶信息，列车状态信息，列车关注状态信息，列车动态信息，列车计划运行图，列车时刻信息，首末班车信息，子线路列车信息，逻辑轨道基础信息，投影坐标，线路坐标偏移信息，目的地信息，TOS 动态信息，手动 TOS 信息，线路下车站名称映射，主变电站基础信息
行车信息业务通信服务	获取列车信息，获取在线列车信息，获取实时在线列车信息，获取实时晚点列车信息，清除长时间不移动列车，设置列车状态，获取列车状态，设置列车关注状态，获取列车关注信息，根据车次号获取计划列车运行时刻表，获取晚点列车排名，获取指定实时偏移列车信息，获取逻辑轨道信息，获取基准坐标偏移信息，获取列车目的地信息，获取列车基础信息，获取列车下一到达站编号，获取车站实时行车间隔时间
客流信息业务通信服务	获取单独实体的实时 TOS 信息，获取线路实时 TOS 信息，刷新手动实时 TOS 信息，刷新已限流车站，刷新已关站车站，获取 5 min 客流进站、出站客流信息，获取 5 min 客流换乘信息，获取前一日 OD 客流信息
电力信息业务通信服务	获取车站电力类型，根据电力类型获取车站信息，获取全部主变电站信息，获取指定主变电站信息，根据车站编号获取车站一次图菜单资源，根据车站编号获取车站 UPS 菜单资源，根据车站编号获取车站模拟量资源
告警信息业务通信服务	订阅告警动态信息，添加告警记录，更新告警记录，确认告警记录，删除告警记录，获取未完成的告警记录，告警信息录入历史库

4.2 容错机制、服务发现、负载均衡设计

上一节分析了 gRPC 通信方案的特点和适用方案,但是对于高价值的分布式应用系统,其本身存在容错性、服务发现、负载均衡的应用缺陷。针对以上应用缺陷,对分布式 gRPC 数据通信的地铁线网指挥中心平台做了如下改进设计:

1) 容错性的改进措施。在该分布式 gRPC 指挥中心平台容错性方面采用了退避重试的算法进行重试。

在某些情况下,有时因为网络故障以及其他原因,在 gRPC 的服务请求中未能得到预期的响应,重试是解决分布式系统容错性经常采用的措施。在分布式系统中,微服务系统重试可能会触发多个其他请求或重试操作,并导致级联效应。为应用程序和客户端添加定时轮询重试逻辑,将产生大量的重试操作,可能会使网络情况变得更糟糕,甚至会阻止应用程序恢复。为减少重试带来的影响,应该减少重试的数量,并使用指数退避算法来持续增加 gRPC 重试之间的延迟时间,直到达到最大限制。退避重试可以使 gRPC 请求在等待一定时间后再发送,等待时间是随指数增长,从而避免频繁的触发冲突。

2) 服务发现的改进措施。在 gRPC 通信管理服务里(类似于简易的 zookeeper 功能)开发了相关注册服务信息、心跳保活机制功能来分别用于服务注册和健康检查,从而实现了简易的服务发现功能。客户端通过向 gRPC 通信管理服务发送客户端连接信息进行服务注册, gRPC 通信管理服务负责集中调配对应的 gRPC 通信业务服务进行链接。通过使用 gRPC 长链接的方式, gRPC 通信管理服务向 gRPC 通信业务服务发送心跳,如果长期没有响应则会将其剔除,并

及时释放掉失去保活性的业务服务资源,以实现健康检查功能。

3) 负载均衡的改进措施。采用 Nginx 实现负载均衡的设计, Nginx 版本要求: 1.13.10+。Nginx 1.13.10 新增了对 gRPC 的原生支持。gRPC 必须使用 HTTP/2 传输数据,支持明文和 TLS 加密数据,支持流式数据交互,可以充分展现 HTTP/2 连接的多路复用和流式特性。使用指令 `grpc_pass` 来指定代理的 gRPC 服务器地址。地址前缀协议有两种:“`grpc: //`”是与 gRPC 服务器端以明文的方式交互、“`grpcs: //`”是与 gRPC 服务器端以 TLS 加密方式交互。

5 分布式 gRPC 地铁线网指挥中心平台测试、应用与改进

5.1 测试用例

gRPC 的核心协议是 Protocol Buffer, 目前广泛应用于系统之间的数据交互协议还有 Json、XML。其中 Json 是 XML 的简化、优化协议, Json 的各项性能指标都要高于 XML。现只对 Protocol Buffer 与 Json 进行比较。编写了两个测试用例, 分别用来测试 Json 与 Protocol Buffer 的编解码速度、协议容量大小。测试用例都是使用相同结构的 Java 类(用户: 姓名、年龄、电话)进行测试的。Json 与 Protocol Buffer 测试用例对比如表 3 所示。

1) 测编解码速度。从表 3 可知, 转换次数在 1 000 以上, Protocol Buffer 的编解码性能比 Json 高出很多; 次数在 10 万以上, Protocol Buffer 的编解码性能则远远高出 Json 的性能。内存占用方面, Json 占用内存到达 106 字节, 而 Protocol Buffer 的占用内存是 34 字节, Protocol Buffer 的内存占用只有 Json 的 1/3。

表 3 Json 与 Protocol Buffer 测试用例对比

Table 3 Comparison between the test cases of Jason and Protocol Buffer

测试用例	用例描述	测次手段	Json	Protocol Buffer
测编解码速度	同样结构体的 Java 类, 并且将 Java 对象互相转换成 Json 数据、Protocol Buffer 数据进行比较	转换 100 次	内存: 106 字节 编码: 32 ms, 解码: 8 ms	内存: 34 字节 编码: 32 ms, 解码: 4 ms
		转换 1 000 次	内存: 106 字节 编码: 39 ms, 解码: 21 ms	内存: 34 字节 编码: 37 ms, 解码: 8 ms
		转换 10 000 次	内存: 106 字节 编码: 126 ms, 解码: 93 ms	内存: 34 字节 编码: 49 ms, 解码: 23 ms
		转换 100 000 次	内存: 106 字节 编码: 248 ms, 解码: 180 ms	内存: 34 字节 编码: 51 ms, 解码: 58 ms
测协议容量大小	同样结构体的 Java 类对象存储成 Json 数据、Protocol Buffer 数据进行比较	存储 10 000 个对象	322 K	117 K
		存储 100 000 个对象	3.14 M	1.14 M
		存储 1 000 000 个对象	31.4 M	11.4 M
		存储 50 000 000 个对象	157 M	57.2 M

2) 测协议容量大小。从表 3 可知, 由于 Protocol Buffer 是采用二进制进行序列化, 压缩效率高, 存储相同数量对象的容量大小是 Json 的 1/3。

5.2 应用与改进

地铁线网指挥中心平台系统组成比较复杂, 数据交互要求高, 表 4 为地铁线网指挥中心平台系统组成与数据交互简表。在天津 ETC、上海市 COCC 地铁线网指挥中心平台的建设和实现中, 有如下几点改进的应用成果:

1) 对于行车、客流、电力、告警管理、ISCS

监视等实时性要求高的系统, 使用 gRPC 分布式通信可有效保证数据一直处于长链接传输中, 可以减少服务之间频繁建立链接的资源开销, 可用低延时、高并发、高刷新率的方式同时传递到同一或者不同操作系统平台的多客户端。在实际地铁线网指挥中心项目开发中已经实现与应用, 以上海为例(2021 年 12 月 30 日前, 20 条运营线路、508 座车站、列车保有量突破 7 000 辆), 对线网行车(500 ms 访问频率)、客流(5 min 访问频率)、ISCS 监视等实时数据进行传输。

表 4 地铁线网指挥中心平台系统组成与数据交互

Table 4 Platform system composition and data interaction of a metro network control center

分类	指挥平台	语言	程序架构	部署系统	数据抽取方式	数据交互周期
应用层	综合监视系统	C#	CS、BS 混合	windows	请求、流式数据	按需、实时
	数据应用	java web	BS	centos	请求、流式数据	按需、实时
	运营执行	java web	BS	centos	请求、流式数据	按需、实时
	共享门户、APP	andriod	CS、BS 混合	andriod、centos	请求、流式数据	按需、实时
	大屏可视化	java web	BS	centos	请求、流式数据	按需、实时
	信息组团	C#	CS	windows	请求	按需
服务层	行车计算服务群	C#	CS	windows server	流式数据	实时(500 ms)
	客流计算服务群	C#	CS	windows server	流式数据	实时(5 min)
	程序更新服务	C#	CS	windows server	流式数据	实时
	报警处置服务	java	CS	centos	流式数据	实时
	报警计算服务群	java	CS	centos	流式数据	实时
	数据提取服务群	java	CS	centos	流式数据	实时
	应急算法服务	asp.net	BS	windows server	请求	按需
	视屏分析服务	C++	CS	windows server	流式数据	实时

2) 对于 CCTV 视频文件, 综合监控设备图纸文件以及设备实时数据 XML、Json 文件等非结构化数据, 支持流式数据传输。而不需要采用自行封装的 Socket、FTP 等方式进行传输。

3) 线网指挥中心平台包含的业务系统较多, 涉及 C/S 架构程序和 B/S 架构程序混合集成, 并且涉及 C#、C++、java web、asp.net、andriod 等多语言、多框架体系的程序系统集成^[11], 如表 4。gRPC 可以适应多语言、跨平台环境系统的集成, 可以解决该问题。客户端和服务端可以分别使用 gPRC 支持的不同语言实现。

4) 由于线网指挥中心平台需要接入各专业子系统的设备数据, 需要有高吞吐、大数据量数据进行传输通信, 分布式 gRPC 通信中心所需网络转换协议 Protocol Buffer 数据协议, 其容量小, 占用带宽少, 可以进行高效的网络传输。参考表 3 Json 与 Protocol

Buffer 测试用例对比表。

5) 线网指挥中心平台可以简单方便地规范多业务系统之间的数据格式。一次定义, 多次使用, 使各系统间业务数据格式保持一致。例如: 应急管理系统需要调用综合监视系统的告警信息触发相应的应急联动和应急预案, 则可以定义一次告警信息协议, 多业务系统之间可以无缝调用。

6) 分布式 gRPC 通信层的实现, 使得业务应用层的众多系统, 可以不用直接访问集成服务层的众多计算服务群以及基础平台层的数据库平台、分布式存储平台、大数据储存平台和消息中间件平台, 使得访问入口统一。

6 结语

综上所述, 本文设计并实现了一种基于 gRPC 分布式数据通信中心的地铁线网指挥中心平台架构。本

文 gRPC 分布式数据通信中心的实现,提升了线网行车、客流、ISCS 监视等实时数据系统和非实时数据业务系统的传输效率;可以适应线网指挥中心多语言平台、多框架开发体系的融合;并能够实现 proto 数据协议一次定义,多系统间可以无缝读取调用;另外,使得线网指挥中心平台服务访问路口统一,可以协调多服务之间的调用。在天津 ETC、上海 COCC 线网指挥中心平台的开发建设中, gRPC 分布式数据通信中心经过多年的尝试、改善和应用,使得地铁线网指挥中心平台能够稳定运行,为地铁线网指挥中心平台的数据通信进一步优化发展提供可靠的保证。

参考文献

- [1] 徐杰, 贾利民, 秦勇, 等. 城市轨道交通综合监控平台系统集成的研究[J]. 铁道学报, 2007, 29(3): 107-112.
XU Jie, JIA Limin, QIN Yong, et al. Study on integrated monitor and control system for urban rail transit[J]. Journal of the China railway society, 2007, 29(3): 107-112.
- [2] 林湛. 智能城轨总体框架研究[J]. 铁路计算机应用, 2020, 29(11): 1-8.
LIN Zhan. Overall framework of intelligent urban rail transit[J]. Railway computer application, 2020, 29(11): 1-8.
- [3] 张铭, 王富章, 李平. 城市轨道交通网络化运营辅助决策与应急平台[J]. 中国铁道科学, 2012, 33(1): 113-120.
ZHANG Ming, WANG Fuzhang, LI Ping. The platform of network operation assistant decision making and emergency for urban rail transit[J]. China railway science, 2012, 33(1): 113-120.
- [4] 赵瑜颢. 基于 Grpc 的分布式远程过程调用框架设计与开发[J]. 现代信息技术, 2021, 5(4): 88-92.
ZHAO Yuhao. Design and development of distributed remote procedure call framework based on grpc[J]. Modern information technology, 2021, 5(4): 88-92.
- [5] 刘小磊, 程伟华, 戚林成. 基于 gRPC 协议的监控调用链中组件性能指标研究[J]. 自动化技术与应用, 2021, 40(8): 81-84.
LIU Xiaolei, CHENG Weihua, QI Lincheng. Research on component performance index in call chain monitoring based on grpc protocol[J]. Techniques of automation and applications, 2021, 40(8): 81-84.
- [6] 汤钦华, 侯晋, 王路遥, 等. 基于远程过程调用的网络流量可视化技术研究与应[J]. 中国数字医学, 2020, 15(4): 117-120.
TANG Qinhua, HOU Jin, WANG Luyao, et al. Research and implementation of telemetry technology with gRPC[J]. China digital medicine, 2020, 15(4): 117-120.
- [7] 梁强升. 城市轨道交通线网运营管理指挥中心建设与管理方案研究[J]. 都市快轨交通, 2020, 33(1): 127-133.
LIANG Qiangsheng. Construction and management of urban rail transit network operations management command center[J]. Urban rapid rail transit, 2020, 33(1): 127-133.
- [8] 穆怀远. 地铁综合监控系统关键技术研究与应用[J]. 数字技术与应用, 2017(5): 90-91.
MU Huaiyuan. Research and application of key technologies in metro integrated monitoring system[J]. Digital technology and application, 2017(5): 90-91.
- [9] 朱莉莉. 地铁临时线网应急指挥中心客流及行车设备监控平台设计方案[J]. 铁路计算机应用, 2016, 25(7): 53-56.
ZHU Lili. Passenger flow and train operation equipment monitoring platform for emergency command center in metro temporary line[J]. Railway computer application, 2016, 25(7): 53-56.
- [10] 张浩. 地铁线网指挥中心系统功能探讨[J]. 科技视界, 2017(25): 189-190.
ZHANG Hao. Discussion on the function of subway network command center system[J]. Science & technology vision, 2017(25): 189-190.
- [11] 白丽, 王石生, 姚湘静, 等. 城市轨道交通综合智能运维平台研究与设计[J]. 铁路计算机应用, 2020, 29(11): 62-65.
BAI Li, WANG Shisheng, YAO Xiangjing, et al. Integrated intelligent operation and maintenance platform for urban rail transit[J]. Railway computer application, 2020, 29(11): 62-65.

(编辑: 王艳菊)