

面向轨道交通管理系统的实时数据共享平台技术

刘超 李申 苏成 王伟 黄勃

(交控科技股份有限公司, 100070, 北京)

摘要 [目的]随着轨道交通行业的发展,各种传感、定位技术的进步,许多低成本的数据收集设备连接到互联网。访问丰富的业务运营数据成为可能,但信息过载也为数据管理带来了挑战。高效的实时数据处理能力能够为决策系统提供有用的信息,支撑轨道交通行业的各种人工智能算法。针对此问题,对面向轨道交通管理系统的实时数据共享平台技术进行了研究。[方法]分析了共享平台在轨道交通行业中的作用,明确其在提升运营效率和安全性方面的重要性;梳理了业务系统对实时数据共享的主要需求,并根据需求整理出技术选择标准;通过定性和定量的比较,评估不同中间件的适用性,以确定最适合平台实现的中间件。[结果及结论]试验结果表明,基于内存数据网格的中间件 Hazelcast 软件在处理大量并发请求和高数据负载时表现出色,且其响应时间增长稳定,适合作为轨道交通管理系统实时数据共享平台的技术基础。并且 Hazelcast 软件具有高性能、易配置以及强大的集群支持能力,能够满足轨道交通管理系统对实时数据处理的需求。

关键词 轨道交通管理系统;实时数据共享平台;内存数据网格;Hazelcast 软件

中图分类号 F530.7

DOI:10.16037/j.1007-869x.2025.03.057

Real-time Data Sharing Platform Technology for Rail Transit Management Systems

LIU Chao, LI Shen, SU Cheng, WANG Wei, HUANG Qing

(Traffic Control Technology Co., Ltd., 100070, Beijing, China)

Abstract [Objective] With the development of the rail transit industry and advancements in sensing and positioning technologies, numerous low-cost data collection devices are now connected to the internet, which enables access to abundant operational data but also poses challenges in data management due to information overload. Efficient real-time data processing capabilities can provide actionable insights for decision-making systems and support various artificial intelligence algorithms used in rail transit sector. Addressing this issue, the real-time data sharing platform technology for rail transit management systems is investigated. [Method] The role of sharing

platforms in the rail transit industry is analyzed, emphasizing their importance in improving operational efficiency and safety. The primary requirements for real-time data sharing from business systems are outlined and the technical selection criteria are derived from these requirements. Finally, a qualitative and quantitative comparison is conducted to evaluate the suitability of different middleware options, identifying the most appropriate middleware for platform implementation. [Result & Conclusion] Experimental results indicate that Hazelcast, a middleware based on an in-memory data grid, performs excellently well in handling large volumes of concurrent requests and high data loads, with stable response time growth. This makes it a suitable technical foundation of real-time data sharing platform for rail transit management systems. Additionally, Hazelcast demonstrates high performance, ease of configuration, and robust cluster support capabilities, effectively meeting the real-time data processing requirements of rail transit management systems.

Key words rail transit management system; real-time data sharing platform; in-memory data grid; Hazelcast software

随着城市轨道交通建设规模的逐年扩大和客运量的持续增长,轨道交通发展正面临诸多挑战。交通管理系统,通过整合计算机技术、通信技术和系统集成技术,实现了对列车运行逻辑的集中管理和实时优化,进而提升了运输能力、增强了安全性能、降低了运营成本并提高了运营效率^[1]。然而,当前的交通管理系统在数据利用方面尚存不足,各业务系统产生的多源异构数据相互独立^[2],这在一定程度上限制了交通管理能力的提升。

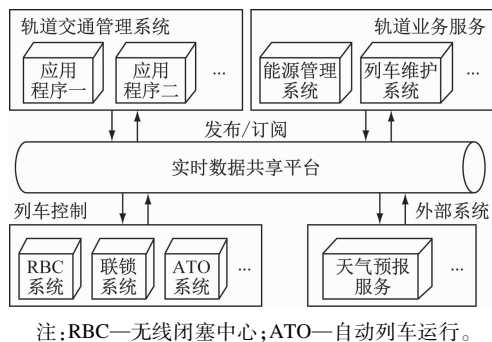
本研究聚焦于实时数据共享技术,旨在开发一个面向轨道交通领域的数据共享平台,该平台能够将内部和外部服务连接到交通管理系统的标准接口上,从而促进实时数据的动态交换。

首先,本文对面向交通管理系统的实时数据共享概念进行了阐述;接着,分析了轨道交通业务对平台的需求,并提出了初步的实现方案;然后,梳理

了基于中间件实现数据共享的技术选择标准;最后,对基于内存技术的中间件 Redis(远程字典服务器)和 Hazelcast 软件在实现共享方案中的可行性进行了评估。

1 相关概念

实时数据共享平台集成了外部系统和轨道交通管理系统内部应用,通过统一的平台和标准接口,为服务提供实时数据管理、访问及通信功能。实时数据共享架构如图 1 所示。



注:RBC—无线闭塞中心;ATO—自动列车运行。

图 1 实时数据共享架构

Fig. 1 Architecture of real-time data sharing

1.1 轨道交通管理系统

轨道交通管理系统是一个旨在提高轨道交通运营自动化程度的综合管理系统,它覆盖的业务范围比当前的行车调度指挥系统更广。该系统应用围绕实时数据构建,接收实时数据输入,并根据相应算法进行预测、验证并给出决策意见。传统的轨道交通管理系统由单一供应商提供专有接口,每个业务程序独立托管在自己的平台上。为了打破这种隔离,轨道交通管理系统服务通过应用框架进行统一管理^[3],并利用实时数据共享平台进行业务之间的通信。

1.2 其他系统

实时数据共享平台除了为轨道交通管理系统应用提供共享数据和通信环境外,也支持与轨道业务服务及外部系统的通信。

1.3 数据共享模式

在数据共享模式方面,企业服务总线作为传统中间件技术与 XML、Web 服务等技术结合的产物^[4],为企业服务集成架构的运行提供了基础设施,并为系统调用提供了统一的服务调用接口;作为一种成熟的架构,已在多个领域广泛应用^[5-6]。

另一种以数据为中心的中间件模式则以统一

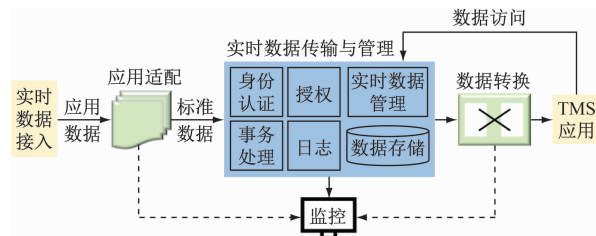
的、全局的数据模型传输数据,像数据库一样将已知的结构强加给传输的数据。服务可以通过类似增加、读取、更新、删除的操作访问数据,应用之间相互解耦,这种访问一致性简化了系统之间的数据共享。

2 实现方案与技术评估

本节根据轨道交通管理系统对实时数据共享的需求提出了参考解决方案,并梳理了实现数据共享平台的技术选择标准。接着,根据这些标准对基于内存的代表性中间件进行了评估。

2.1 实现方案

面向轨道交通管理系统的实时数据共享平台旨在根据不同内部和外部客户端的需求,在统一的通信基础架构中为不同客户提供轨道交通运营中涉及的各种资产的“现在”和“预测”状态数据。实现方案包括将应用接入实时数据共享平台、提供平台服务、为轨道交通管理系统应用提供所需的实时数据及提供流程监控功能等。实时数据单向接入访问流程如图 2 所示。



注:TMS—任务管理系统。

图 2 实时数据单向接入访问流程

Fig. 2 Process of real-time data one-way connection and access

2.1.1 应用适配

在应用适配方面,实时数据共享平台需要与不同类型的应用进行通信,这些应用通常基于专有的数据模型建立,对同一个实体对象有着不同的理解和表示。因此,需要实现与内部、外部应用的适配方案。在平台服务方面,为了保证数据的安全接入和访问,平台需要提供权限认证、授权、日志、数据存储、数据传递等服务。

2.1.2 平台服务

中间件技术常用于构建分布式系统,屏蔽底层异构平台和系统,简化应用之间的数据传输,实现跨平台的信息传输和交换。同时,中间件也提供权限验证、日志等功能。因此,下文对中间件技术应

用于面向轨道交通实时数据共享平台的可行性进行了评估。

在技术选择标准方面,实现数据共享平台的中间件需要满足轨道交通管理系统对实时数据共享的需求。这包括数据传递方式、数据访问、消息和消息管理、队列和队列管理以及事务等方面的要求。

为了保证数据的安全接入和访问,平台需要为用户和交互系统提供权限认证服务,用于客户端操作权限管理的授权服务、事务支持、日志访问功能以及数据传递、存储管理功能。

中间件技术常用构建分布式系统,屏蔽底层异构平台和系统,简化应用之间的数据传输,实现跨平台的信息传输和交换。同时也提供权限验证、日志等功能。因此下文对中间件技术应用于面向轨道交通实时数据共享平台的可行性进行评估。

2.2 技术选择标准

实现数据共享平台的中间件需要满足轨道交通管理系统对实时数据共享的需求。具体包括如下内容。

1) 数据传递方式。实时数据共享平台需要支持三种消息传输方式,如图3所示。发布订阅模式消费者根据主题的数据结构进行订阅。共享平台利用此模式管理列车交通信息分发类的需求。点对点模式确保只有一个接收者能接收到文档或者消息。用于发送增加列车或者人员信息给实时交通计划服务这类直接通信的场景。请求应答可以看作是点对点传输方式中的一种,要求在程序发送消息后能获得接收方的响应。

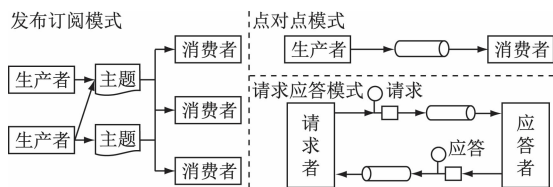


图3 数据传递模式

Fig.3 Data delivery mode

2) 数据访问。轨道交通管理系统业务产生的数据不限于字符串、数字等基本类型,也包括对象、视频等数据,一些数据需要进行序列化才能够存储和转发,中间件需要支持二进制序列化的内容读写,以满足各种业务数据的访问需求。同时需要对主题的访问深度进行配置,允许历史信息的访问。

3) 消息和消息管理。消息管理机制提供对消

息生命周期的有效管理。中间件应支持简单和复杂的数据类型,以满足轨道交通业务中命令、文档、事件等丰富的消息传递需求;轨道交通管理系统预测服务根据列车的当前状态和基础设施信息对运行列车的行为进行预测。

2.3 中间件比较分析

将数据加载到内存中,可以极大地提高读写性能,因此内存技术是实现实时数据共享的重要手段。下面根据2.2节中提出的标准评估两个典型的基于内存技术的中间件,为实现面向轨道交通管理系统的实时数据共享提供参考。

2.3.1 基于内存的中间件

Redis是基于内存技术的代表性中间件,是一个以键和值的形式存储数据的开源单线程内存缓存,支持异步事件循环处理。Redis 5.0推出了流模型通过RDB(远程字典服务器数据库快照)和AOF(追加只读文件)和AOF^[7]持久化策略来尽量保证消息可靠。Hazelcast IMDG(内存数据网格)是一个开源分布式计算的内存数据存储平台,提供了基于JVM(Java编程语言虚拟机)的分布式数据存储和计算环境^[8]。节点之间共享内存数据和计算负载。与Redis最大的区别在于Hazelcast IMDG在内存存储之上还能应用程序提供分布式计算,利用集体CPU的处理能力,当应用程序将任务提交给服务器时,每个实例只需访问运行实例的服务器上的内存数据,不需要通过网络传输数据。

2.3.2 定性比较的结果与分析

根据梳理的需求对Redis和Hazelcast IMDG进行对比的结果如表1所示,其中支持表示中间件的功能可以覆盖共享平台对该内容的需求,部分支持表示只能实现该内容的部分功能。从表1中可以看出基于内存技术的Redis和Hazelcast可以覆盖大部分面向轨道交通管理系统对实时数据共享的需求。

在数据传输方面,两者都支持点对点、发布订阅的传输方式。Redis协议位于TCP(传输控制协议)层之上,支持串行和双工的请求响应模式,发布数据时将消息推送到每个订阅者的连接缓冲区。如果单个消息超过订阅者的缓存容量,没有及时消费导致数据积累过多都会造成Redis出错,虽然新增了缓存限制,但大数据量的消息推送仍会造成内存资源紧张。Hazelcast IMDG提供数据分发机制,发布订阅操作是在集群范围内的,当集群中的某个成员注册了一个新的主题时,该节点会根据集群注

册表向所有集群中的成员发送一条消息,同步主题状态。当新的成员加入时,也会同步集群中所有注

册过的主题。Hazelcast 本身不支持请求/响应模式,但可以通过两个队列进行模拟。

表 1 功能对比
Tab. 1 Comparison of features

需求类型	需求	Redis	Hazelcast IMDG
数据传输方式	点对点	支持	支持
	发布-订阅	支持	支持
	请求-应答	支持	部分支持
数据访问	允许读、写二进制序列化形式的单个键/值对	支持	部分支持
	允许读、写相同类型的键/值对组	支持	部分支持
	键/值变化通知	支持	支持
	主题历史记录访问	不支持	不支持
	支持类 SQL 语言	不支持	支持
	简单和复杂的数据类型	支持	支持
	自描述和结构化数据格式,如 XML	支持	
消息和消息管理	消息过期	支持	支持
	消息压缩	支持	部分支持
	消息优先级	不支持	不支持
	消息选择器	部分支持	支持
队列和队列管理	使用消息队列存储和转发消息	不支持	支持
	消息持久性	部分支持	部分支持
	消息的完整性	不支持	支持
事务	支持事务	支持	支持
性能	高吞吐量	支持	支持
	低延迟	支持	支持
	伸缩性	支持	支持
	稳定性	支持	支持
使用	易于部署	支持	支持
	兼容性	支持	支持

在数据访问方面,两者都允许应用程序和模块访问(读或写)二进制序列化形式的单个键/值对内容和相同类型的键/值对组,Redis 通过管道命令将给定键/值对的值更新(序列化数据的更改)通知给模块。Hazelcast 通过监听器侦听事件。两者都允许显示地或者利用过期机制删除数据。

在消息和队列管理方面,两者都支持消息过期,但都无法处理消息优先级,在实际应用需要使用自定义的逻辑进行弥补。Redis 没有基于消息内容路由消息的能力。Hazelcast 提供消息过滤支持,并为了保证消息的可靠性,支持创建可靠主题,使用 Ringbuffer(环形缓冲区)数据结构备份主题事

件。由于数据存在内存中,两者都无法提供完全的消息持久化。

在使用方面,两个中间件都具有平台无关性,能够在虚拟环境中运行。Hazelcast 更易于部署,并能根据部署策略、使用模式选择客户端/服务器或者嵌入式体系结构的方式,通过动态的添加和删除服务器实现弹性的内存资源。

在性能方面,与 Redis 集群模式向上扩展的方式不同,Hazelcast 具有服务自动发现能力,通过简单的配置可以方便地添加和删除节点,实现弹性扩展。在有节点加入或者离开集群时所有节点更新分区表,自动实现数据平衡。而 Redis 没有服务自

动发现能力,很难在自定义的云部署中使用。

2.3.3 定量比较结果与分析

为轨道交通管理系统业务提供近缓存性能的数据访问能力以及高效处理多设备、系统数据接入的能力是实现数据共享平台的关键,本节对上述两种中间件在这两方面的性能做进一步的试验分析。

2.3.3.1 试验方法与配置

1) 试验设备及环境配置。使用具有相同配置的3台装有Linux操作系统的服务器构建,详细配置信息如表2所示。其中Redis采用主从集群的部署模式(3主3从),Hazelcast采用3个节点的分布式集群部署模式,并使用内嵌模式和客户端/服务器模式两种部署策略进行测试。

表2 试验环境

Tab.2 Experiment environment

名称	配置
系统版本	Linux version 3.10.0-1160.el7.x86_64
CPU	Intel(R) Xeon(R) Gold 6252N CPU @ 2.30 GHz 4核
RAM	24G
JDK	1.8
Redis 版本	5.0.0
Hazelcast 版本	5.13

2) 试验数据。模拟轨道交通数据,随机生成2000万条数据作为试验数据集,每条测试数据的键由UUID(Universally Unique Identifier,通用唯一标识符)生成,值为20位随机字符串(由大小写字母和数字组成)。

3) 试验方法。使用Java编程语言编写代码,利用生成的数据对Redis和Hazelcast IMDG进行插入、读取、删除的性能试验。试验包含两个部分,第一部分对Redis和Hazelcast IMDG的数据插入、读取、删除性能进行测试。计算在单线程的条件下,两者完成同等数据量操作的耗时。第二部分试验测试两个中间件在相同线程下的吞吐量,线程的数目从1增加到100。计算在单位时间内,不同线程条件下处理请求的个数。

2.3.3.2 时延试验

时延试验涵盖了插入、读取和删除3种数据操作类型,操作时延定义为从客户端初始化至操作线程执行完毕的时间间隔。数据操作时延结果如图4所示,横轴代表数据量,范围从1万条至2000万

条,纵轴表示操作时延。时延越低,中间件性能越优。图4a)为插入操作的结果,表明在插入100万条数据时三者性能相近,但随着数据量增加,Redis的插入速度略胜一筹。图4b)为读取操作的结果,随着数据量增长,Hazelcast在数据访问方面的优势愈发显著,特别是在数据量为2000万条时,内嵌模式的Hazelcast仅用1.73 s,客户端/服务器模式用时4.595 s,性能达到Redis的3倍以上。图c)呈现了删除操作的结果,三者删除性能差异不大。

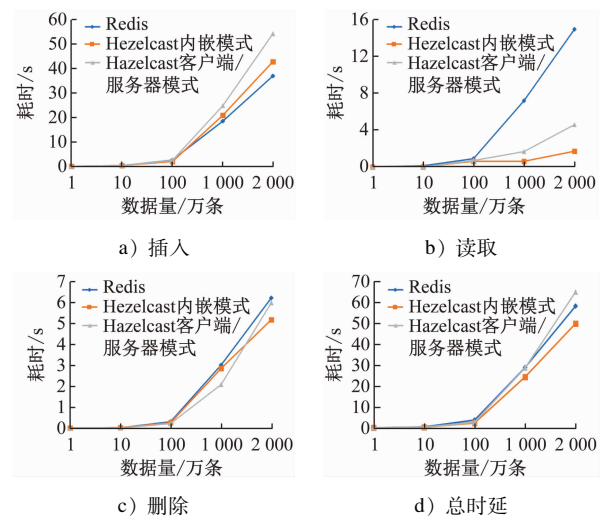


图4 数据操作时延结果

Fig. 4 Latency results of data operation

2.3.3.3 吞吐量试验

吞吐量试验评估了3000万个请求在不同线程数量下的处理能力。试验过程中,所有数据被均匀分配至与线程数量相等的组,每组数据由相应线程并行处理,所有线程完成后统计结果。吞吐量定义为单位时间内处理的请求数量,吞吐量越高,性能越优。吞吐量试验结果如图5所示,随着线程数量增加,吞吐量相应提升,能够满足多系统、多组件数据接入的处理需求。其中,Hazelcast内嵌模式相较于Redis展现出更高的吞吐量。

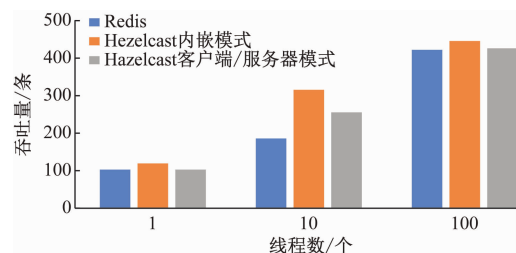


图5 吞吐量试验结果

Fig. 5 Throughput test results

两部分试验的结果均表明,在数据访问方面,Hazelcast IMDG(无论是内嵌模式还是客户端/服务器模式)相较于 Redis 具有显著优势,更适合作为实时数据共享平台的中间件。在吞吐量测试中,Hazelcast 同样展现出相对于 Redis 的优势。

3 结语

本文研究了面向轨道交通管理系统的实时数据共享技术,重点评估了基于内存技术的中间件(特别是 Hazelcast IMDG)在实现轨道交通实时数据共享方面的可行性。通过定性和定量试验分析,验证了 Hazelcast IMDG 能够作为面向轨道交通管理系统的实时数据共享平台的有效解决方案,且相较于广泛应用的 Redis 具有明显优势。

参考文献

- [1] 毛保华. 城市轨道交通系统运营管理[M]. 北京:人民交通出版社,2006.
MAO Baohua. Operations and management for urban rail transit [M]. Beijing: China Communications Press, 2006.
- [2] 赵新勇. 基于多源异构数据的高速公路交通安全评估方法[D]. 哈尔滨:哈尔滨工业大学,2013.
ZHAO Xinyong. Highway traffic safety assessment method based on multi-source heterogeneous data[D]. Harbin: Harbin Institute of Technology, 2013.
- [3] 叶涛. 基于 Dubbo 服务的分布式轨道交通管理系统设计与实现[D]. 北京:中国科学院大学,2020.
YE Tao. Design and implementation of distributed rail transit man-

agement system based on Dubbo service[D]. Beijing: University of Chinese Academy of Sciences, 2020.

- [4] 邵欢庆,康建初. 企业服务总线的应用[J]. 计算机工程,2007,33(2):220.
SHAO Huanqing, KANG Jianchu. Research and application of enterprise service bus [J]. Computer Engineering, 2007, 33(2): 220.
- [5] 郭瑜. 基于 ESB 构架的交通行业数据交换平台的应用研究[J]. 计算机系统应用,2008,17(9):83.
GUO Yu. Research on application of ESB-based transportation data exchange system[J]. Computer Systems & Applications, 2008, 17(9): 83.
- [6] AZIZ O, FAROOQ M S, ABID A, et al. Research trends in enterprise service bus (ESB) applications: a systematic mapping study[J]. IEEE Access, 2020, 8: 31180.
- [7] XU M, XU X, XU J, et al. A forensic analysis method for Redis database based on RDB and AOF file[J]. Journal of Computers, 2014, 9(11): 2538.
- [8] SALHI H, ODEH F, NASSER R, et al. Open source in-memory data grid systems: benchmarking Hazelcast and Infinispan[C]//Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering. L' Aquila Italy. New York: ACM, 2017: 163.

• 收稿日期:2022-12-08 修回日期:2023-03-14 出版日期:2025-03-10
Received:2022-12-08 Revised:2023-03-14 Published:2025-03-10
• 通信作者:刘超,高级工程师, chao.liu@bj-tet.com
• ©《城市轨道交通研究》杂志社,开放获取 CC BY-NC-ND 协议
© Urban Mass Transit Magazine Press. This is an open access article under the CC BY-NC-ND license

(上接第 302 页)

- ZHANG Peng, WANG Jian, WU Juan, et al. Research and application of key mobile payment technologies in the Nanjing Metro [J]. Urban Rapid Rail Transit, 2020, 33(6): 146.
- [2] 陶克. 人脸识别技术在城市轨道交通自动售检票系统中的应用[J]. 城市轨道交通研究, 2022, 25(10): 198.
TAO Ke. Application of face recognition technology in urban rail transit AFC system [J]. Urban Mass Transit, 2022, 25(10): 198.
- [3] 杨静,韩丽东. 人工智能双目视觉闸机的可靠性分析[J]. 机械工业标准化与质量,2020(2):41.
YANG Jing, HAN Lidong. Reliability analysis of stereo vision based artificial intelligent access control gate[J]. Machinery Industry Standardization & Quality,2020(2):41.
- [4] 钟尖. 基于手机信令的综合交通枢纽客流监测技术研究[D]. 重庆:重庆交通大学,2017.

ZHONG Jian. Research on passenger flow monitoring technology of integrated transportation hub based on mobile phone signaling [D]. Chongqing: Chongqing Jiaotong University, 2017.

- [5] 丁敬安,张欣海. 基于手机信令数据的地铁乘客路径识别研究[J]. 中国电子科学研究院学报,2019,14(11):1194.
DING Jing'an, ZHANG Xinhai. Research on subway passenger path identification based on mobile phone signaling data[J]. Journal of China Academy of Electronics and Information Technology, 2019, 14(11): 1194.

• 收稿日期:2022-12-30 修回日期:2023-02-12 出版日期:2025-03-10
Received:2022-12-30 Revised:2023-02-12 Published:2025-03-10
• 通信作者:杨俊义,高级工程师,327979080@qq.com
• ©《城市轨道交通研究》杂志社,开放获取 CC BY-NC-ND 协议
© Urban Mass Transit Magazine Press. This is an open access article under the CC BY-NC-ND license