董隶鋆, 卿粼波, 冯田,等. 基于 Flink 的智慧社区数据分析系统设计与实现[J]. 智能计算机与应用,2025,15(2):52-58.
DOI:10.20169/j.issn.2095-2163.24091101

基于 Flink 的智慧社区数据分析系统设计与实现

董隶鋆[1], 卿粼波[1], 冯 田[2], 毛建国[1], 赵 炜[2]
(1 四川大学 电子信息学院, 成都 610065; 2 四川大学 建筑与环境学院, 成都 610000)

摘 要: 本文设计了一种基于 Apache Flink 的智慧社区数据分析系统,该系统涵盖数据集成、存储、处理及应用四大层次,以满足智慧社区对多源数据高效处理的需求。该系统充分发挥 Flink 的分布式处理与流处理优势,对接并高效处理来自社区的多样化数据,包括结构化、半结构化及非结构化数据,实现数据的实时转换与加载,确保了数据的时效性与准确性;数据处理核心环节,系统深度融合 Flink 的 ETL(Extract, Transform, Load)处理方法,不仅为社区管理者构建了一套闭环式的数据处理过程,还提供了一套完整的实时数据分析解决方案。实验验证了该系统在处理复杂数据任务中的可靠性与实用性,为智慧社区的管理和服务提供了有力支持。

关键词: Apache Flink; 数据分析; 智慧社区; ETL 处理
中图分类号: TP391.4 **文献标志码**: A **文章编号**: 2095-2163(2025)02-0052-07

Design and implementation of a smart community data analysis system based on Flink

DONG Lijun[1], QING Linbo[1], FENG Tian[2], MAO Jianguo[1], ZHAO Wei[2]
(1 College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China;
2 College of Architecture and Environment, Sichuan University, Chengdu 610000, China)

Abstract: This article proposes a smart community data analysis system based on Apache Flink, which covers four levels of data integration, storage, processing, and application to meet the efficient processing needs of smart communities for multi-source data. This system fully leverages Flink's advantages in distributed processing and stream processing, and efficiently processes diverse data from the community, including structured, semi-structured, and unstructured data, achieving real-time conversion and loading of data, ensuring data timeliness and accuracy. The core link of data processing, the deep integration of Flink's ETL (Extract, Transform, Load) processing method, not only builds a closed-loop data processing process for community managers, but also provides a complete real-time data analysis solution. Experimental verification has shown the reliability and practicality of the system in handling complex data tasks, providing strong support for the management and service of smart communities.

Key words: Apache Flink; data analysis; smart community; ETL processing

0 引 言

在智慧社区建设的征程中,数据不仅是宝贵的资源,更是实现精准管理的核心[1]。社区数据源众多,包括政府网格信息、居民个人档案、社区服务机器人数据以及社区小程序和网络爬虫收集的在线民情与社区新闻。这些数据涵盖公共服务多个层面,反映了管理方式与社区分析方法的多元化。

在社区"微网实格"智慧服务平台建设中,多源数据的高效整合是核心挑战[2]。由于各部门信息独立,形成"信息孤岛",导致数据采集效率低且实时性差,还存在数据协同差和风险高的问题[3-4]。传统的数据处理方法多基于 MapReduce 分布式批处理框架、Storm 实时流处理系统或 Spark 分布式计

基金项目: 成都市重点研发支撑计划(2023-YF09-00019-SN)。
作者简介: 董隶鋆(1999—),男,硕士研究生,主要研究方向:图像处理和信息论,人工智能与计算机视觉; 冯 田(1995—),女,博士,硕士生导师,主要研究方向:低碳城市,低碳交通; 毛建国(2000—),男,硕士研究生,主要研究方向:图像处理和信息论,人工智能与计算机视觉; 赵 炜(1974—),男,博士,教授,博士生导师,主要研究方向:城市设计与城市更新,历史城镇与乡村规划。
通信作者: 卿粼波(1982—),男,博士,教授,博士生导师,主要研究方向:人工智能与计算机视觉,图像/视频处理。Email: qing_lb@scu.edu.cn。
收稿日期: 2024-09-11

算引擎。MapReduce 主要适用于大规模数据的批处理，但其处理延迟较高，不适合实时流处理[5]；Storm 专注于低延迟的流数据处理，但在处理复杂的批量任务时表现较差[6]；Spark 提供了更高效的批处理和流处理能力，但在处理多源异构数据时仍面临一定挑战[7]。因此，高效采集和处理不同来源与格式的数据成为关键。

在智慧社区数据分析中，离线数据分析侧重于对历史数据进行深入挖掘，揭示长期趋势与模式[8]；而实时数据分析则着眼于捕捉即时事件，提供快速响应与决策支持[9]。针对上述挑战，本文设计了一种基于 Flink 的智慧社区数据分析系统。该系统不仅提出了面向社区数据的分析流程与方法，而且细化了面向实时数据的分析方法。利用 Flink 的流批一体处理优势，灵活应对从静态历史数据集到实时数据流的处理需求，实施有效的 ETL（Extract，Transform，Load）操作。综合离线数据分析和实时数据分析两个分支，智慧社区数据分析系统，智慧社区数据分析系统为社区管理提供了全方位的数据支撑，提高系统实时性和精准性，使社区治理更智能化。

1　智慧社区数据分析系统总体框架

智慧社区数据分析系统总体框架如图 1 所示，主要分为 4 层：数据集成层、数据存储层、数据处理层和数据应用层。

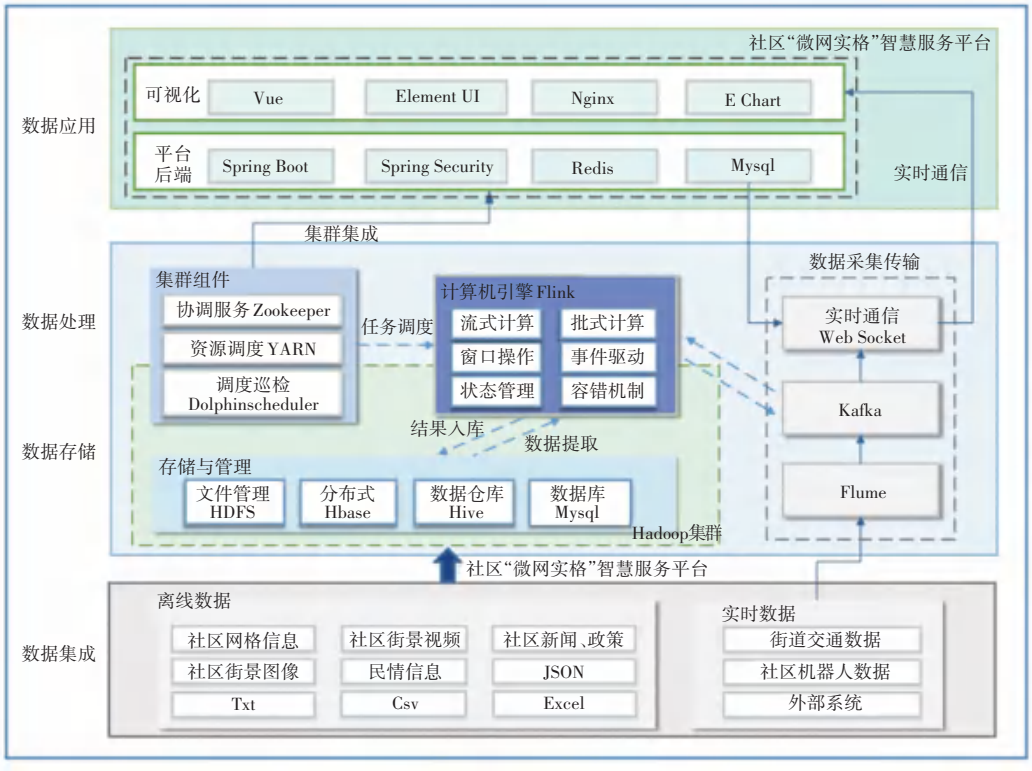

图 1　智慧社区数据分析系统总体框架

Fig. 1　Overall framework of smart community data analysis system

1.1　数据集成层

数据集成层针对多样化和多源性的社区数据进行分类整合。离线数据广泛采集于社区的各种数据源，通常来源于“微网实格”智慧服务平台的积极采集和上传。而实时数据则涵盖社区的即时动态，比如街道交通数据、社区机器人数据等，这些数据通常通过外部系统借助 Flume 等数据传输框架进行实时采集。

Apache Flume 是一个高效且具备容错能力的分布式数据收集系统，广泛用于高效地收集、聚合和移动大量的数据，尤其适用于智慧社区中各类数据的实时采集[10]。Flume 基础架构如图 2 所示。

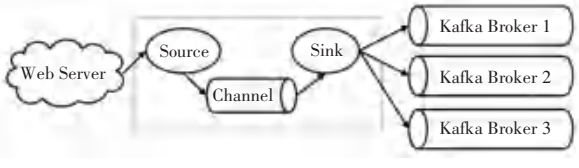

图 2　Flume 基础架构

Fig. 2　Flume Infrastructure

Flume 的工作原理基于定义 Source、Channel 和 Sink 3 个核心组件来实现数据的收集和传输[11]。

Source 是数据采集的起点,负责从智慧社区的各个数据源(如传感器、智能设备)捕获数据;Channel 临时存储,确保数据在传输中的安全性和可靠性;Sink 则是数据的终点,负责将收集到的数据送往指定的存储系统,例如(Hadoop Distributed File System)或 Kafka 消息队列系统。

1.2 数据存储层

数据存储层是数据持久化的关键所在。这一层使用了 HDFS、HBase(Hadoop DataBase)、HIVE(Hadoop Integrated View Environment)、MySQL 等多种存储技术,为海量数据提供高吞吐量的存储能力,确保存储着社区中产生的各类数据;提供了快速读写大量数据的能力,为数据处理层提供所需的原始数据支撑,同时也存储着数据处理的最终结果。

系统架构基于稳固的 Hadoop 集群,集群通过 Zookeeper 分布式协调服务进行协调,并借助 YARN(Yet Another Resource Negotiator)资源调度平台实现资源和作业调度[12]。为保证社区实时数据流的高效处理,系统使用 Kafka 消息队列系统作为中间件数据管道,确保数据传输的高吞吐量和伸缩性[13]。

1.3 数据处理层

数据处理层采用高性能的 Flink 流处理框架,支持流批一体,提供优秀的状态管理、容错机制和灵活的事件驱动处理能力[14]。

整个数据分析过程以社区“微网实格”智慧服务平台、分布式文件系统 HDFS、关系型数据库 Mysql 这三大核心为基础。平台集成了大数据相关技术组件,结合基于 Flink 计算引擎的 ETL 模块,为社区管理者提供一套完整的处理闭环,以便有效管理和分析大量社区数据。社区数据处理过程如图 3 所示。

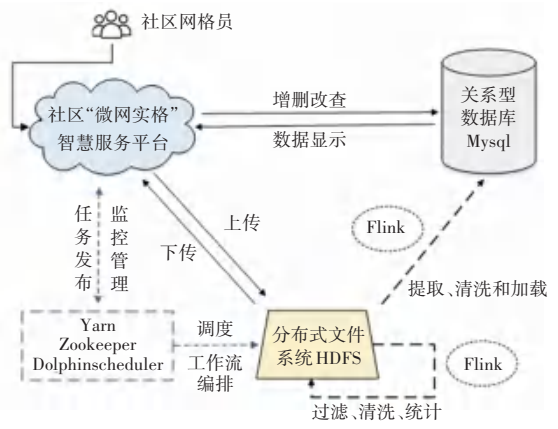

图 3 社区数据处理过程

Fig. 3 Community data processing

1)针对 HDFS 中的文本文件,利用 Flink 框架的高效处理能力,网格员可以执行 ETL 流程,确保数据质量,并安全地将结果存回 HDFS;

2)对于社区网格员上传的多样化文件(如 .CSV、.Excel、.txt 等),Flink 框架同样执行 ETL 流程,然后将其映射至关系型数据库,优化数据存储,增强智慧服务平台的数据展示和管理能力[15];

3)针对时效性更高的实时数据流,数据处理流程更加依赖于自动化运行[16]。社区网格员主要通过平台,以及 DolphinScheduler、Yarn 等组件进行任务流的监控与管理。

实时数据与离线数据在多个方面存在本质的差异[17]。

基于这种区分,针对社区实时数据分析的过程如图 4 所示。这一过程旨在从数据的源头开始,即时捕获、处理数据流,经 Flume 采集、Kafka 传输、Flink 高速计算,至 WebSocket 与 ECharts 可视化,各环节无缝对接,构建高效自动化分析体系。同时,引入数据持久化存储,保障数据安全与长期价值。

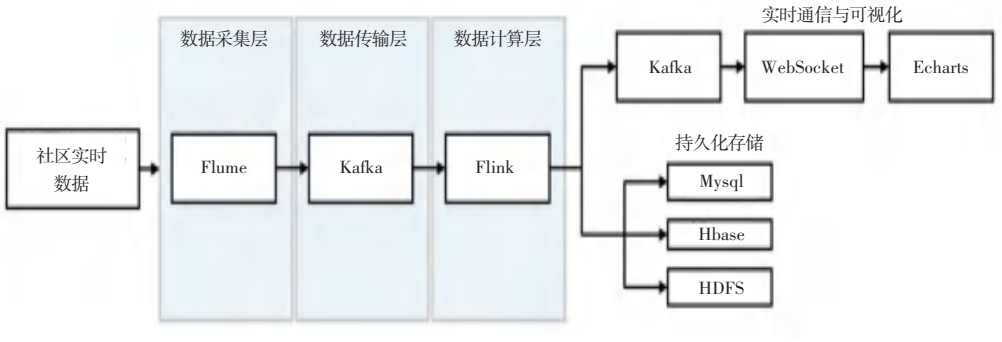

图 4 社区实时数据分析过程

Fig. 4 Real time community data analysis process

在数据采集层,Flume 通过 Kafka Sink 配置,利用 Kafka 分区策略,将多源数据按规则分散到 Kafka 不同分区,优化数据处理效率,确保来自社区不同设备和传感器的数据被正确分类和分配,优化数据处

理和分析的效率[18]。

Flume 支持动态调整 Kafka 主题，允许根据实时监测到的数据流量和模式灵活调整分区数及主题配置[19]。比如，当某些社区事件导致数据洪峰时，Flume 可以动态增加分区以应对突增的数据量，确保系统的稳定运行[20]。

1.4　数据应用层

数据应用层依托“微网实格”智慧服务平台，结合 SpringBoot 微服务框架、SpringSecurity 安全框架、Redis 分布式缓存系统及 Hadoop 大数据处理平台，构建全面的大数据应用方案。前端采用 Vue. js 前端框架、Echarts 数据可视化库实现直观的数据可视化和大屏展示，WebSocket 实时双向通信协议支持双向通信，实时推送社区数据。ECharts 可视化库转化复杂数据为生动图表，帮助管理者和居民掌握社区动态。

该系统架构分离清晰，前端专注业务逻辑与交互，后端负责高效数据传输与处理。设计分离关注点，提升可扩展性、可靠性和维护性，前端专注 UI（User Interface）与逻辑，后端保障数据传输效率与稳定。

2　基于 Flink 的社区数据 ETL 处理方法

Apache Flink 的易用性和高性能，使其成为执行社区数据 ETL 任务的理想选择。

Flink 为数据处理提供了两个主要的 API：DataSet API 和 DataStream API。DataSet API 是针对批处理的标准选择，专为有界数据集设计，数据集大小有上限且在处理之前已经完全收集完毕，能够在处理开始前应用全局优化，如排序和去重。然而，随着 Flink 1. 12 版本的推出，DataSet API 已被整合到 DataStream API 中，旨在统一批处理和流处理。Flink 分层 API 如图 5 所示。

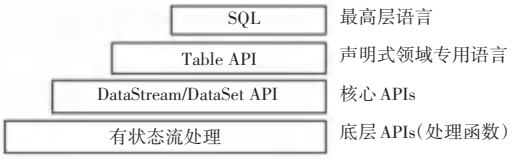

图 5　Flink 的分层 API

Fig. 5　Flink′s Layered API

DataStream API 支持无界及有界数据处理，低延迟、高灵活，成为实时数据流与静态数据处理的优选方案[21]

针对社区数据处理的 3 个关键环节，本文提出了不同的数据处理思路，并分别应用了相关的 Flink 算子配置以优化数据处理性能。

2.1　数据清洗

静态批处理主要聚焦于对存储在 HDFS 中的已有数据集进行一次性的大规模处理。

首先，利用 readTextFile（inputPath）方法读取 HDFS 中的文本数据源，通过调用 distinct（）函数对数据集中的重复元素进行去重；其次，使用 writeAsText（outputPath）方法将去重后的数据集重写回 HDFS，保证重写时覆盖。

然而，在处理大规模数据集时，distinct（）方法可能会引起性能瓶颈。因此需要审慎评估其适用性。这种方法要求系统在内存中维护一个包含所有唯一元素的集合，如果数据量庞大，可能导致内存耗尽，可能会引起性能瓶颈。为优化这一过程，可以使用 KeyedStream 算子配合状态管理和窗口操作进行去重。

动态流处理则侧重于处理持续产生的数据流，并能即时响应数据更新。首先使用 fromSource 算子从数据源接收实时数据流；其次，通过 flatMap 算子处理数据流中的每一行文本，将其转换为二元组格式，其中第一个字段作为去重的键，第二字段保留原始文本数据。

使用 keyBy 算子对数据进行分组，并为每个键分组设置时间窗口。在这个固定时间窗口中，实现去重逻辑，每个时间窗口只接收并输出第一个具有独特键的数据记录，从而在相同时间内相同键的数据只被输出一次，实现了基于时间的去重处理。最后，通过将处理后的数据流写入 HDFS，完成实时数据流的 ETL 操作。Flink 数据去重任务链如图 6 所示。

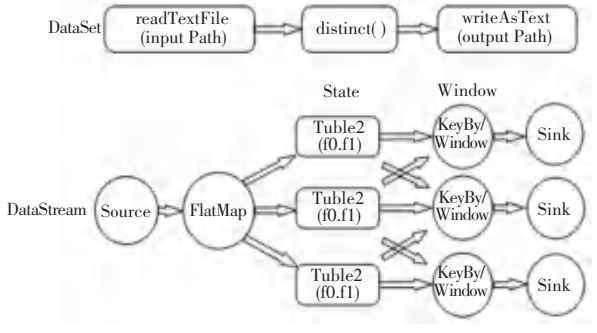

图 6　Flink 数据去重任务链示意图

Fig. 6　Flink data deduplication task chain diagram

2.2　数据转换

通过 Flink 平台上实施数据流处理并写入 MySQL 数据库的流程是数据管道设计的另一个关键部分，涉及将网格员上传到 HDFS 中的离线数据

或已持久化存储到 HDFS 中的实时数据读取,经过处理,有效地存储至关系型数据库中。Flink 数据过滤与数据库写入任务链如图 7 所示。

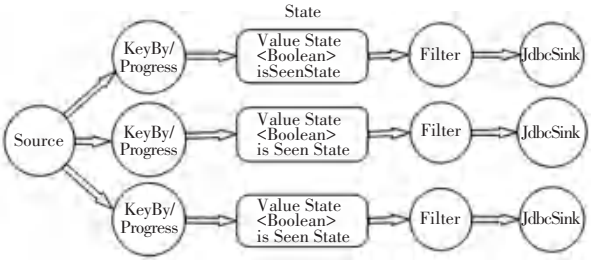

图 7 Flink 数据过滤与数据库写入任务链示意图

Fig. 7 Flink data filtering and database write task chain diagram

首先,从数据源中读取实时数据流,确保能够捕获到连续产生的数据,并为后续的处理流程打下基础;其次,将字符串数据映射到数据库表的相关字段,这一转换步骤是实现从原始数据结构到数据库结构转化的关键。

映射转换后,根据数据中某个唯一字段(比如,实体的 ID)对数据流进行分流。这个过程中,每个独特的键(ID)对应的数据将会分配至不同的任务,便于并行处理,标志每个唯一字段(ID)的数据是否之前已被处理过。

基于这个状态,执行去重操作,保证了每个独特键(ID)的数据只会被处理一次,实现了按键(ID)的去重。

此外,为了进一步保证数据质量,检查数据流中的记录,剔除那些包含空值字段的条目,确保向数据库中传输的数据具备完整性。

处理好的数据通过写入 MySQL 数据库,保障在遇到暂时性写入失败时,系统可以进行的重试次数,指定每个批次包含的记录条数;优化数据批处理的性能,设定了批处理的时间窗口。

2.3 实时数据处理任务

由于社区实时数据的时效性,实时数据流的 ETL 处理过程不依赖社区网格员的处理任务分发,更多的是任务的监控。Flink 利用 Kafka 的分区策略,使 Flink 按照分区读取数据;Flink 按照 Kafka 分区消费数据,保证了高效的数据处理和分布式的并行性。

在 DataStream API 的使用中,首先连接 Kafka 并创建一个新的数据流源。在数据流中进行数据解析,将原始数据转换为有用的格式;其次,根据特定的关键字段(例如 ID)对数据进行分组,以指定窗口类型和长度来定义数据处理的范围,并计算所需的指标;最后,将处理后的数据写回 Kafka。Flink 实时数据处理任务链如图 8 所示。

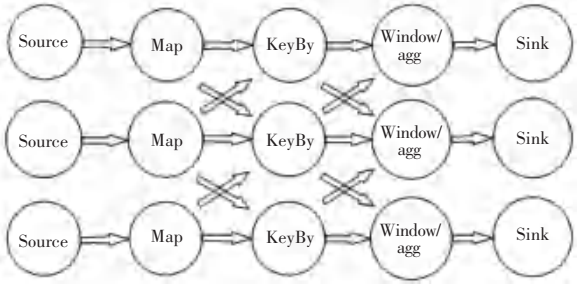

图 8 Flink 实时数据处理任务链

Fig. 8 Flink real-time data processing task chain

3 实验与分析

本文通过仿真对基于 Apache Flink 流式计算框架的社区数据处理方法进行了可行性验证。数据均来自于政府开放数据平台的真实数据,离线数据包括某社区某图书馆文学类书籍藏书信息、某社区新增图书馆馆藏图书清单以及某社区年度电梯运行周期信息。实时数据来源于某区域 2023 年 3 月到 11 月的街道交通数据。

采用 Java 编程语言,编辑器为 IntelliJ IDEA Community 2023。在同一局域网下,使用 3 台服务器节点搭建集群实验环境,3 台服务器的配置信息见表 1。

表 1 服务器配置信息

Table 1 Server configuration information

	操作系统	处理器配置	内存/G
主节点	CentOS Linux 7	Intel(R) Core(TM) i5-4460 CPU @ 3.20 GHz	16
从节点 1	CentOS Linux 7	Intel(R) Xeon(R) CPUE5-2687W v3@ 3.10 GHz	16
从节点 2	Ubuntu 18.04.6 LTS	Intel(R) Xeon(R) CPUE5-2686W v4@ 2.30 GHz	16

利用 Flink 1.17,对上传到 HDFS 中的某图书馆藏书信息和某社区年度电梯周期信息进行数据去重操作,对比了在不同数据量下,DataSet API 的去重方法和 DataStream API 的窗口控制方法的处理性能差异。

在 DataStream API 的使用中,针对图书馆藏书

信息，建立二元组格式，以“条码号”作为去重的键；针对年度电梯周期信息则以上传的具体时间作为唯一键。在相同数据和并行度都为 1 的情况下，处理速率取决于为各键分组的时间窗口的时间界定以及最大乱序间隔，数据去重处理结果见表 2。

表 2　数据去重处理结果

Table 2　Data deduplication results

数据	处理前/条	方法	时间窗口/s	最大乱序时间间隔/s	处理后/条	速率/(条/s)
书籍藏书信息	5 005	DataSet			4 994	1 200
电梯运行信息	185 910	DataSet			180 500	19 000
书籍藏书信息	5 005	DataStreamSource	5	3	4 994	1 900
			10	5		
电梯运行信息	185 910	DataStreamSource	5	3	185 633	19 000
			10	5		

根据数据去重处理的结果分析，关于 DataSet 和 DataStream 处理离线数据集的性能问题，取决于数据的处理方式和具体场景。对于少量数据，特别是批处理场景中，DataSet API 可能在处理有界数据时表现出更好的性能，尤其在旧版的 Flink 中，这是因为 DataSet API 可以利用整个数据集的先验知识进行全局优化；对于大数据集，特别是类似社区年度电梯周期信息的情况，需要在内存中维护一个包含所有唯一元素“上传时间”的集合，不如 DataStream API 流处理的高效性。

在使用 DataStream API 的窗口方法对电梯运行数据进行清洗时，清洗效果不佳的情况，通常是由于时间窗口的时间界定和最大乱序时间间隔设置不合理引起的，当将时间窗口增加到 10 s，最大乱序时间间隔增加到 5 s 后，清洗结果变得准确。因此，采用窗口控制时，窗口大小以及允许乱序时间需要根据数据实际的乱序程度进行调整。

本文利用 Apache Flink 对存储在 HDFS 中的某社区新增图书馆馆藏图书清单进行了数据清洗与入库(至 MySQL 数据库)的操作。为了评估不同配置对处理性能的影响，本文在保持实验环境、处理数据及并行度一致的前提下，调整了 JDBCSink 的参数配置，以匹配不同的写入模式需求，具体包括重试次数、批次大小和最大等待时间，得到的结果见表 3。

表 3　清洗入库结果

Table 3　Cleaning and storage results

数据	处理前/条	需求	JDBCSink 参数	处理后/条	处理速率/(条/s)
新增图书清单	10 000	高效	3	9 256	3 500
			1 000		
			0		
	10 000	平衡	2	9 256	900
			500		
			5 000		
	10 000	低延迟	1	9 256	1 500
			100		
			1 000		

由表 3 可见，Flink 对数据清洗并写入 MySQL 的 ETL 操作，数据批量写入数据库的性能受到 JDBCSink 配置的影响。由于 JDBCSink 涉及频繁的网络交互，配置参数时需要参考数据量、数据库性能、网络稳定性、数据紧急程度等因素。

通过对 70 条街道 2023 年 3 月份到 11 月份的 500 万条交通数据，获取包含上传时间戳字段和时间片字段的数据，根据街道 ID 写入 Kafka 的不同分区，并利用 Flink 进行分析任务计算。本文选择以下统计指标进行分析：样本总行程时间(s)、平均行程车速(km/h)、样本总行驶长度(m)、交通拥堵指数等，前 10 条街道在几个月期间的交通拥堵指数趋势如图 9 所示。交通拥堵指数越大，表示越拥堵。交通指数划分为 5 个等级：0-2 表示道路畅通，2-4 表示基本畅通，4-6 表示车辆缓行，6-8 代表交通拥堵，8-10 代表极其拥堵。

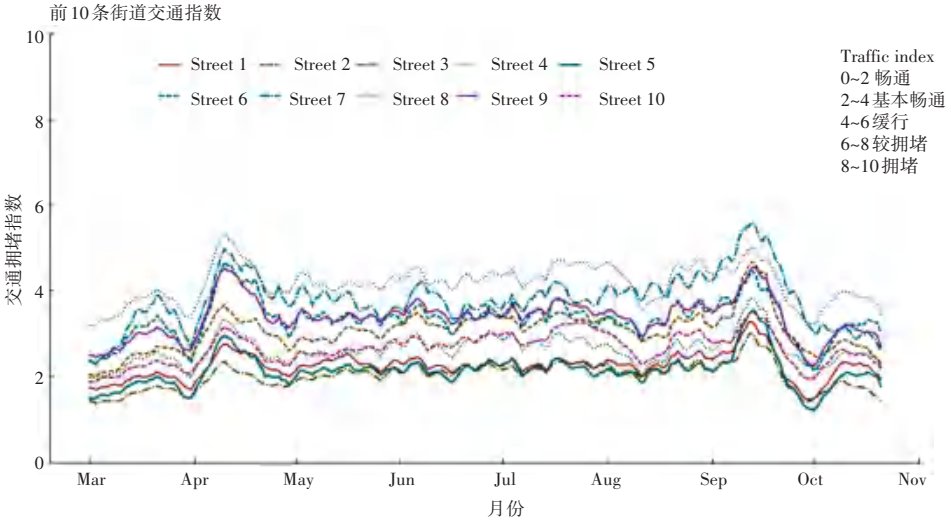

图 9 街道交通拥堵趋势

Fig. 9 Trend chart of street traffic congestion

实验结果验证了本文设计的基于 Flink 的智慧社区数据分析系统在离线数据处理和实时数据处理方面的有效性。通过对图书馆藏书信息和社区年度电梯周期信息进行数据清洗操作,说明了离线数据处理流程与方法中不同参数配置对处理性能的影响;对街道交通实时数据的多维度分析证实了该系统在处理复杂数据任务中的可靠性和实用性。

4 结束语

本文设计了一种基于 Flink 的智慧社区数据分析系统,通过设计和验证一系列离线和实时数据处理方法,展示了系统在处理复杂社区数据任务中的卓越性和实用性。通过实验证明了系统总体框架设计的有效性,包括社区离线数据处理流程和实时数据分析流程。该系统不仅提升了社区数据综合应用的效率,还对智慧社区的升级和建设提供了有力支持。这一研究成果为未来智慧社区的发展提供了坚实的基础,期望能够进一步推动大数据技术在社区管理和服务中的广泛应用,使得社区治理更高效、更智能。

参考文献

[1] 张艳国,朱士涛. 大数据融入智慧社区建设:时代价值与现实路径[J]. 江汉论坛,2021,521(11):117-124.
[2] 张淳瑞. 面向智慧园区的数据治理平台设计与实现[D]. 北京:北方工业大学,2024.
[3] 牛庆丽,朱耀琴. 基于 Spark 计算的大数据终端潜在异常识别仿真[J]. 计算机仿真,2024,41(1):518-521.
[4] 何健. 基于 MapReduce 平台的大数据查询与处理优化算法[J]. 电脑编程技巧与维护,2024,467(5):107-109.
[5] 赵娟,程国钟. 基于 Hadoop、Storm、Samza、Spark 及 Flink 大数据处理框架的比较研究[J]. 信息系统工程,2017(6):117-119.
[6] 宋灵城. Flink 和 Spark Streaming 流式计算模型比较分析[J]. 通信技术,2020,53(1):59-62.
[7] 代明竹,高嵩峰. 基于 Hadoop、Spark 及 Flink 大规模数据分析的性能评价[J]. 中国电子科学研究院学报,2018,13(2):149-155.
[8] 蒋雷,白伟丽,李小红. 基于 ClickHouse 的实时数据仓库的基础架构研究[J]. 现代计算机,2024,30(11):91-95.
[9] 王仁杰. 基于 Flink 的实时数据仓库优化技术研究[D]. 成都:电子科技大学,2023.
[10] 王诗航,张明昊,赵运哲,等. 基于 Flume 和 Flink 的轨道交通智能运维数据接收解析系统[J]. 技术与市场,2023,30(4):86-90.
[11] VYAS S, TYAGI R K, JAIN C, et al. Performance evaluation of apache kafka - a modern platform for real time data streaming [C]//Proceedings of the 2nd International Conference on Innovative Practices in Technology and Management. Piscataway, NJ: IEEE, 2022:465-470.
[12] 王思明. 基于 Kafka 系统的数据中心网络流量调度方案设计[D]. 成都:电子科技大学,2024.
[13] GARG N. Apache Kafka [M]. Birmingham, UK: Packt Publishing, 2013.
[14] 梁方玮,薛涛. 面向物流服务的海量日志实时流处理平台[J]. 计算机系统应用,2021,30(10):68-75.
[15] CARBONE P, KATSIFODIMOS A, EWEN S, et al. Apache flink: Stream and batch processing in a single engine[J]. The Bulletin of the Technical Committee on Data Engineering, 2015, 38(4):1.
[16] 刘辉,陈刚. 基于 Flink 的工业大数据实时分析平台[J]. 电子技术与软件工程,2021(6):185-187.
[17] 曹云柯. 一种基于 Flink 实时数仓的系统设计及功能实现研究[J]. 电子技术与软件工程,2022,224(6):219-222.
[18] KATSIFODIMOS A, SCHELTER S. Apache flink: Stream analytics at scale[C]//Proceedings of the 2016 IEEE International Conference on Cloud Engineering Workshop. Piscataway, NJ: IEEE, 2016: 193-193.
[19] 傅宣登,吴志林. Apache Flink 复杂事件处理语言的形式语义[J]. 软件学报,2024,35(10):4510-4532.
[20] 赵润发,娄渊胜,叶枫,等. 基于 Flink 的工业大数据平台研究与应用[J]. 计算机工程与设计,2022,43(3):886-894.
[21] 高立京,陈志敏,王春梅,等. 基于 Flink 的空间科学卫星数据实时处理方法[J]. 计算机仿真,2023,40(7):26-31.