计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(5):67−76 [doi: 10.15888/j.cnki.csa.009105]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

场站飞行保障数据可视化平台设计与应用①

刘君阳[1], 朱世松[2]

[1](空军勤务学院 研究生大队, 徐州 221000)
[2](空军勤务学院 飞行保障指挥系, 徐州 221000)
通信作者: 刘君阳, E-mail: 1746618275@qq.com

摘　要: 为提升航空兵场站信息化建设过程中的数据应用与管理能力, 提出了一种基于 Spark 的场站飞行保障大数据可视化平台. 以场站信息化系统和物联网络数据采集数据为基础, 利用 Spark 计算引擎集成 Kafka 消息队列, 使用 Hive 完成数据列表库的建立和存储, 基于 Spark RDD 和 Spark SQL 完成数据预处理与交互, 并选择 Vue 框架嵌入 ECharts 组件完成前端数据可视化呈现, 并最终对设计方案进行了实现与应用. 相较于当前业务隔离的信息系统建设模式, 平台具备更高的数据融合与处理分析能力, 能够更好地实现场站飞行保障数据价值.

关键词: 飞行保障; 大数据; 大数据处理平台; 数据可视化; Spark; 态势感知; 数据分析

引用格式: 刘君阳,朱世松.场站飞行保障数据可视化平台设计与应用.计算机系统应用,2023,32(5):67–76. http://www.c-s-a.org.cn/1003-3254/9105.html

Design and Application of Visual Data Platform for Station Flight Support

LIU Jun-Yang[1], ZHU Shi-Song[2]

[1](Postgraduate Brigade, Air Force Logistics Academy, Xuzhou 221000, China)
[2](Flight Support and Command Department, Air Force Logistics Academy, Xuzhou 221000, China)

Abstract: For better data application and management capability of air-force stations in information construction, this study proposes a Spark-based visualization platform for flight security big data at the stations. On the basis of the data collected with the station informatization system and Internet of Things (IoT) data, a Spark computing engine is used to integrate Kafka message queues, and Hive is used for the establishment and storage of a data list library. Data pre-processing and interaction are completed on the basis of Spark RDD and Spark SQL, and the Vue framework is selected to embed ECharts components to visualize front-end data display. Finally, the design solution is implemented and applied. Compared with the current business-isolated information-system construction mode, the platform has higher abilities for data fusion and processing analysis and can better realize the value of flight security data at the stations.

Key words: flight support; big data; big data processing platform; data visualization; Spark; situation awareness; data analysis

军事数据平台的研究已取得了诸多成果, 文献 [1] 基于数据挖掘技术设计了海战场态势可视化平台架构; 文献 [2] 研究的战区战场环境信息保障平台统筹建立了战场环境大数据探测体系与网络化、无线化、远程化的战场环境信息传输体系; 文献 [3] 提出了航空兵场站应用大数据的方法思路, 并在文献 [4] 中构建了场站飞行保障大数据应用系统基本架构.

总体来看, 针对军事及航空兵场站大数据平台的研究受到了广泛关注, 但缺乏技术层面的指导, 功能的设计缺乏实际成果, 数据分析应用缺乏实现能力. 因此

① 基金项目: 军内科研重点项目 (BKJ19C025)
收稿时间: 2022-10-11; 修改时间: 2022-11-18; 采用时间: 2023-01-06; csa 在线出版时间: 2023-03-30
CNKI 网络首发时间: 2023-03-30

本文针对场站飞行保障活动设计了基于 Spark 引擎的数据可视化平台, 主要完成以下功能.

(1) 集成 Sqoop+Kafka 的末端采集数据传输方法, 并结合 Hive 完成数据列表库的建立和存储.

(2) 基于 Spark RDD 的数据预处理和 Spark SQL 数据交互功能, 并利用 Vue 框架完成前端数据可视化呈现.

(3) 设计数据分析模型, 并在平台实际应用中实现.

1 场站飞行保障数据可视化平台关键技术

1.1 Spark

Spark 是一种大数据计算框架, Apache 官方对 Spark 的定义是: 通用的大数据快速处理引擎, 与侧重大数据存储 Hadoop 的 MapReduce 和 Hive 引擎相比, Spark Core 用于离线计算的可靠性更高, 且更加适用于大数据的计算. 在飞行保障数据可视化平台采用 Spark 架构的主要原因是: Spark 支持 DAG 任务模型, 计算效率相比于传统 MapReduce 计算框架要高出许多; Spark SQL 易于学习, 使用与运维门槛较低; 能够提供机器学习等算法引擎, 支持飞行保障大数据的数据挖掘活动. 在架构方面, Spark 是基于标准 master-slave 模型的计算引擎, 其具体框架[5] 如图 1 所示.

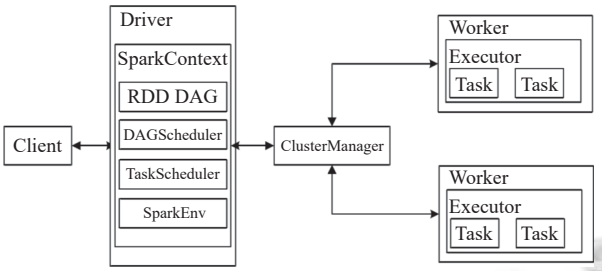

图 1 Spark 基本框架模型

在离线批数据处理方面, 基于 Spark RDD 编写的离线批处理程序运行速度具备明显优势, 且由于数据量的现实条件, 场站层级通常不会出现数十亿级别的数据操作量, Spark 基于内存的计算完全不必担心 OOM 内存溢出等硬性要求问题. 除此之外, Spark SQL 对 JSON 的支持能够契合场站绝大多数信息系统的数据传输接口, 避免了数据整合时的二次开发工作, 且对 ECharts 组件的响应也是考虑的因素之一.

1.2 Kafka

Kafka 是基于 ZooKeeper 协调的分布式消息系统, 最大的特性就是可以实时处理大量数据来满足需求. 虽然目前来看场站飞行保障数据可视化平台对数据时效性的要求并不迫切, 但对数据末端采集活动而言, 传感设备载体 (如装备、设施等) 的状态数据通常以秒为单位进行获取, 定位数据同样要求较高的时间精度和极短的步进宽度, 这就导致持续工作条件下的数据传输层要具备较高的数据传输速率、容错性和可靠性, 从而分担数据采集层的临时存储压力. Kafka 能够以极低硬件配置实现每秒 100k 级别消息的传输, 同时完成数据存储和数据备份; 当 Kafka 在数据传输过程中发生传输节点故障的情况时, 也能够及时释放问题节点内存占用, 以确保数据链路的畅通[6]. 其自带的点对点通信模式和发布-订阅通信模式分别适合完成数据上传与调阅, 不论从结构性、安全性和效率方面来看都十分适合作为平台的数据处理工具.

1.3 Hive

Hive 的定位是数据仓库, 借助于底层 HDFS 或 S3 等对象存储系统完成数据的存储和管理功能, 并支持使用 SQL 语言对数据进行处理和分析, 十分适合批处理模式的数据应用. 同时利用 Spark 引擎替换原有的 MR, 能够极大提升数据处理命令的执行效率. 此外, 当前流行的 Spark+Spark Hive catalog 模式能够使底层 HDFS 中存储的数据在 Spark Hive catalog 所提供的 table-file 映射关系下, 直接使用 Spark 计算引擎提供的 Scala、Java、Python 等 API 或 Spark 语法规范的 SQL 进行处理, 使得原本依托于 Hadoop 架构下 MR 引擎的 Hive 仍能够参与 Spark 计算引擎主导的处理流程中, 进一步简化了数据处理的流程复杂度和难度, 并具备较好的动态可扩展性.

1.4 Vue

Vue 属于渐进式框架, 核心驱动是视图模板引擎, 在此基础上具备构建与连接 Client-Side Routing 等组件完成前端整体框架的搭建, 技术的成熟型也较高, 同时能够兼容 ECharts 数据可视化图表库, 基于浏览器和轻量级的矢量图形库 ZRender 完成前端数据呈现, 并且由于 Vue 多以 HTML、CSS 和 JavaScript 为基础, 故上手难度较低. 此外考虑到场站飞行保障数据的分布广, 响应随机性大等特点, 利用 Vue 的队列自动维护功能可以异步完成事件处理响应循环的缓存, 大幅减少重复操作导致的性能损失[7].

2 场站飞行保障数据可视化平台设计方案

本文旨在设计并实现一款适用于场站飞行保障活动的数据可视化平台, 集成采集、管理、呈现的功能,

为机场保障指挥机构提供态势感知和决策支持能力. 其架构主要分为 5 个组成部分: 数据采集、数据传输、数据处理、数据存储、数据分析与可视化, 从应用层面看, 整体的数据流程结构如图 2 所示.

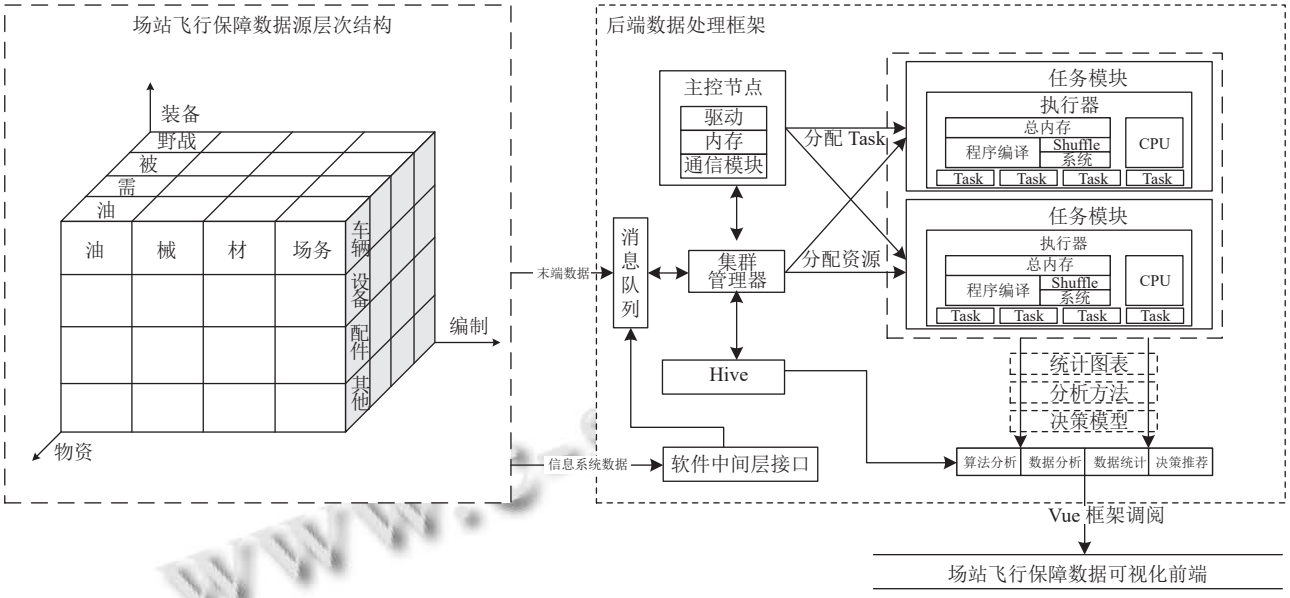

图 2 场站飞行保障数据可视化平台数据流程

2.1 数据采集

获取数据是平台工作流程的第 1 步, 进行采集数据的来源分析能够确保全面而完整地获取平台所需地各类数据, 是平台功能实现的基础. 以目前场站飞行保障活动现状进行分析, 可以将平台数据来源从宏观上划分为信息交易系统数据、机器及传感系统数据和自然环境态势数据 3 类.

(1) 信息交换系统数据

信息交换系统是指场站工作中所涉及和信息化建设完成后将要涉及的, 具有数据交换和分发行为的管理信息系统和智能终端网络设备的集合, 包括但不限于业务部门管理信息系统、外场保障要素信息交互系统、作战指挥层信息交换系统等. 场站飞行保障大数据所需的信息交换系统数据, 是指在以上系统中进行管理的数据或因信息交换行为产生的与飞行保障工作有关的数据集, 如飞行任务与计划中的机种机型、飞行架次、起落时间、物资装备种类、数量需求; 装备种类与数量、保障力量及编制等数据. 该类数据通常采用 Excel、SQL 或 Oracle 等进行存储, 大多能够直接调取应用.

(2) 机器及传感系统数据

机器及传感系统数据来源于来自感应器、量表和其他设施的数据、GPS 定位系统数据等, 如时间空间坐标、人装任务状态 (作业、空闲、准备)、物资数量 (仓储量、在运量、在保量)、设施状态 (温度、湿度、气压、在用情况) 等. 此外还包括功能设备所创建或生成的数据, 例如智能控制器、设施状态和连接物联网的装备数据. 其中来自物联网的数据还应具备构建分析模型的能力, 如连续监测预测性行为 (如当传感器值表示有问题时进行识别), 提供规定的指令 (如警示技术人员在真正出问题之前检查设备) 等. 该类数据通常按照其传感设备工作模式而具有特定的数据格式, 需要经 Sqoop 数据源预先处理, 统一其数据格式后利用 Flume 日志采集手段加入数据队列中.

(3) 态势环境数据

环境态势数据主要由自然态势环境数据和保障态势环境数据组成. 其中自然环境态势数据主要包括 GIS 三维环境数据、BIS 建筑实体数据及气象环境等数据, 主要来源于外部互联网资源; 保障态势环境数据可以从保障任务状态上划分为演训数据 (飞鸟数量及高度、外场要素态势等) 和战时数据 (空地威胁目标速度、高度、打击目标、效果、打击时间、备降迫降机种机型及时间、设施工作状态、备用设施状态、装设备损毁数量) 构成, 主要来源于 GIS 设备和上级指挥信息系统分发. 该类数据的形态为 1、2 类数据的结合,

正常情况下较为固定, 刷新需求较少. 因此在平台中为其应当划分独立的存储管理与分析空间.

2.2 数据传输

机场保障数据类型繁多, 数量庞大, 其数据的传输对传输过程中服务器端的响应时间、传输质量及稳定性、传输信息安全性有很高的要求. 因此选择 Kafka 作为分布式消息队列引擎, 将所有数据在消息队列中进行暂存, 实现数据异步传输和解耦, 并为处于 Spark 框架中的各类数据提供持续性处理能力.

2.3 数据处理

Spark 计算引擎下的批数据处理过程首先是通过从 Kafka 中获取原始数据, 使用 distinct().count() 和 subset() 函数查重选定列数据, 并利用自带的 dropDuplicates 函数进行采集数据集的降重处理, 之后选择 monotonically_increasing_id 进行数据列唯一主键的生成. 完成降重后可以进行空白值处理, 首先利用 miss() 函数查看数据各特征的缺失情况, 过滤无关数据特征和筛选需要填充的数据特征, 之后利用均值、中位数、众数或其他预测方法进行相关信息的填充.

以飞行保障直接相关数据为例, 将 Spark 调取 Kafka 队列中暂存进行数据处理的过程如图 3 所示进行描述.

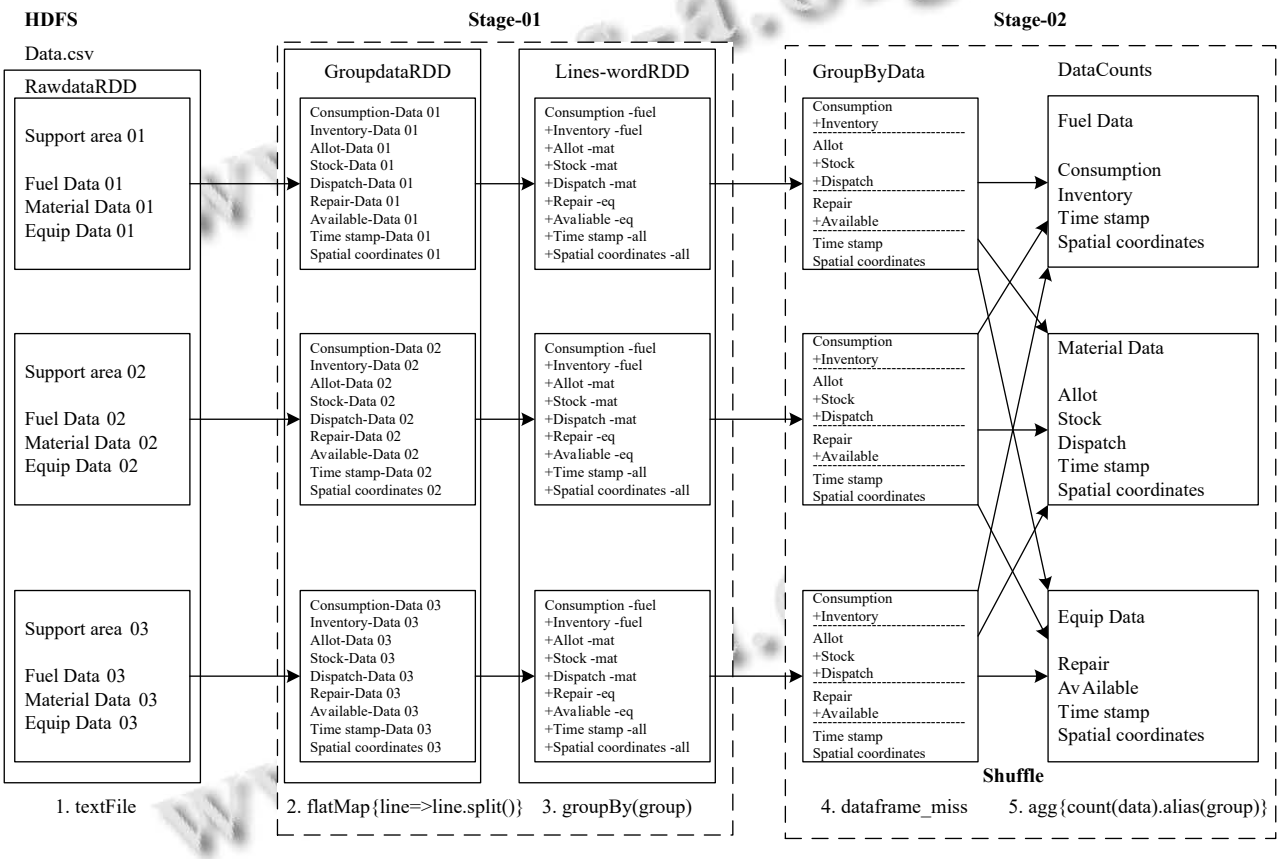

图 3 数据处理流程

(1) 原始数据集 Data.csv 存在于 HDFS 的 3 个 Support area 文件夹中 (此处假设 3 个文件夹分别对应 3 个不同的 HDFS 分布集群机器上), 首先使用 sc.textFile ('Data.csv') 取 HDFS 路径生成 RDD, 数据内容包括各业务领域数据集合, 上述实例以油料、物资、装备末端采集数据为主, 主要内容包括消耗量、库存量、供应量、储备量、派遣情况、维修情况、作业情况等业务数据以及时间戳数据和空间坐标等通用数据, 各文件夹的数据使用区域特征值 01、02、03 进行划分, 分别对应 Support area 01、Support area 02 和 Support area 03.

(2) 使用 textFile.flatMap 将 RDD 中的行数据按照 "\s"拆分为多行数据, 并按照二级表主键划分行数据字符串数组; groupBy() 将每条行数据转换成 data-key 形

式的数据-业务映射元组.

(3) 使用 Spark 中的 Dataframe 对映射元组进行数据清洗. 数据降重使用 dropduplicates 对具有相同数值或相同时间戳的数据进行删除, 由于保障活动处于动态, 故对不同时间戳但其余部分相同的数据不应作为删除对象, 而暂时作为正常数据进行处理; 对缺失值的处理应当采用均值填充思想, 首先使用 miss. rdd. map 对缺失情况进行统计以标识缺失点, 之后利用时间戳组成正常值字典, 利用时间戳数据作为 id 对缺失值进行均值填充, 对于时间戳缺失情况则按照 tresh_value=time_stamp 规则, 使用 dropna 进行行删除. 若连续缺失数量大于 1 条 (tresh=2) 则通过在数据行间新建 DF 暂存上下侧正常数据均值, 并作为临时行间均值计算数据; 异常值处理方面不可随意执行删除函数, 应利用分位数中位数绝对偏差去极值思想对异常元组进行标识, 通过 outliers.filter(data_num).select('time_stamp', 'data_class') 完成异常值选择, 建立新 sheet 将异常数据导入形成异常数据集 outlier_data.csv 进行存储.

(4) 当 dataframe 执行完毕后, 得到了经处理的带有 group 标识的业务数据元组集合, agg().alias() 函数将分组数据按照编制组进行聚合, 并最终在 Kafka 队列中形成 DataCounts 末端数据集合. 将新数据形成数据包发送至数据云资源调度管理器中, 即可供 Spark 引擎作为执行数据分析与挖掘的数据样本.

2.4 数据存储

机场后勤保障数据因其来源途径众多, 数据类型复杂导致数据集异构性较强, 且不同数据的使用场景、时效性、刷新频率等均存在差异. 因此应依据各类数据自身属性特点使用 Hive 作为平台数据仓库, 依托于集群的 HDFS 分布式文件系统通过 Blocks 完成数据的块存储与读取, 同时 Hive 也能够对 Spark 具备良好的支持能力, 能够使用 Spark SQL 进行数据仓库的查询和构建, 且能够响应定制的机场后勤保障指挥功能需求, 对复杂逻辑操作进行 DataFrame、DataSet、RDD 的相互转化, 减少功能逻辑开发的难度. 数据仓库划分为 ODS、DWD、DWS、DWT、ADS 五层结构 [8], 每层通过 SparkSQL 执行各个表的业务逻辑计算. 在本平台中, 信息系统业务数据通过 Sqoop 导入, 末端传感数据通过 Flume 采集, 通过 Kafka+Spark 预处理形成数据仓库的整体原始数据, 通过 Hive 建立 SQL 外部表完成数据关系表构建.

2.5 多源异构数据同步处理

通常飞行保障数据采集完成后的 Kafka 队列同步需要考虑两方面问题, 一是从业务信息交换系统数据库向 Oracle 同步指挥需求数据, 二是从传感及态势数据源向 Hive 同步采集数据.

对于问题一, 以场站某业务部门所用基于 SQL Server 的信息管理系统数据向基于 Oracle 场站飞行保障信息综合物联管理系统同步为例, 可以通过编写脚本, 使用 5 秒级的作业定时调度存储过程对数据进行操作.首先在 SQL Server 端建立通向 Oracle 的链接服务器, 之后在 SQL Server 的同步表中建立 insert、delete、update 和 select 等触发器, 将所需同步数据进行筛选汇总至同一张表 PublishLastRec_SQL 中, 使用游标逐行对该表进行提取, 定时调度频率在本实验中暂定为 30 min, 即每 5 s 对 PublishLastRec_SQL 内数据进行一次更新, 每 30 min 向 Oracle 同步一次 PublishLastRec_SQL 表.

对于问题二, 需要解决的问题是如何从 Kafka 中读取数据并将其写入 Hive 分文件夹中. 此时采集数据临时存储于节点缓存数据库中, 处理方法类似问题一, 通过对日志进行解析并格式化, 向 Kafka 中同步, 但考虑到该类数据的数据量较大且时效性要求高, 脚本执行不能满足所需处理效率, 故选择 Canal 监听节点数据库 binlog 日志并将 binlog 日志信息发送至 Kafka 消息队列, 中间过程基本固定, 重点需要考虑同步后的异构数据管理方法:

(1) 源多条-储单条情况.

(2) 源单条-储单条情况.

(3) 源多条-储多条情况.

对于 (2) 可以通过简单的新增执行语句进行处理, 但对于 (1)、(3), 都有多条主键 key 使储存库中的数据更新的情况, 需要通过业务代码来确保进一步保证数据的一致性.

2.6 数据分析与可视化

数据分析与可视化能够保证为机场后勤保障数据可视化平台的用户呈现直观的数据结果. 前端界面主要完成飞行保障态势展示、末端设备及载体监控并结合后端数据挖掘算法与分析模型对保障成本、保障效能、保障决策、保障流程等方面进行价值分析与技术优化. 因此本平台选取 Apache 基金会旗下开源可视化库 ECharts 作为数据可视化分析工具, 它能够支持多种数据库, 同时还具备多样化的数据展示

能力, 提供直观, 交互丰富, 可高度个性化定制的数据可视化图表. 用户能够按照其个人需求快速进行数据的调阅呈现且避免了与后台的直接接触, 具备优秀的用户交互性能.

2.7 功能集成

首先, 数据源采集活动可采用 Flume 采集机场保障过程中产生的系统日志, 或直接从传感器中获取到数据, 并将这些数据整理为 CSV 格式. 其次, Spark 将整理好的数据发送至 Kafka 消息队列中, 保证数据传输的安全性和顺序性. 然后利用 Spark 获取到暂存至 Kafka 中的数据并对其进行预处理, 去除重复数据、缺失率较高的数据、无效数据等, 处理好后将其存储至 MySQL 中. 可视化分析模块使用在 Vue 框架连接前后端的条件下下使用 ECharts 图标库组件, 实现不同类型数据的可视化呈现.

3 平台实现与性能分析

3.1 测试环境与测试集

本文中的平台集群示例由一个主节点 (master) 和两个子节点 (slave) 搭建而成. 本文中使用 Intel(R) Core i7-10750H 作为主控 CPU, 运行内存为主节点 2 GB, 子节点 1 GB, 存储空间均为 1 TB 硬盘作为测试. 为防止后期数据量达到存储和计算瓶颈, 通过预留后置接口, 确保能够通过动态增加计算机节点来解决问题. 使用 ssh-keygen -t rsa 获取公钥, ssh-copy-id -i hostname 传递公钥, 在目录/HOME/conf/下添加节点以实现免密登录. 考虑到框架相互之间的兼容性与运行质量, 实验数据选择现实采集并进行脱密处理的 15 家场站的多种机型保障活动数据作为呈现, 并结合飞行保障仿真系统平台生成的大量仿真结果作为补充, 总量约为场站正常遂行一个季度训练任务的数据量, 并采用特殊值替换、特征值删除等手段对敏感字段进行处理, 对涉密数据进行了清理. 平台测试所用的具体配置情况如表 1、表 2 所示.

平台数据整体划分为保障活动批数据和末端设备状态数据两种, 保障活动批数据以“保障效能通览-油料保障效能详单”为例, 其数据描述字段为: 种类、计划量、消耗量、适配机型、保障架次、效用比例, 部分数据见表 3. 末端设备状态数据以采集设备状态查询为例, 数据描述字段为 ID、设备类型、设备工作区域、载体、运行状态, 部分数据示例见表 4.

表 1 平台集群硬件配置表

节点名称	IP地址	角色
L-Master	192.168.56.10	主节点
L-Slave-1	192.168.56.11	子节点
L-Slave-2	192.168.56.12	子节点

表 2 平台主要软件及版本

名称	版本
Linux	CentOS 7
Spark	2.7.4
Kafka	2.9.1
ZooKeeper	3.5.9
Vue	2.9.6
ECharts	5.3.2

表 3 油料保障效能详单

种类	计划 (吨)	消耗 (吨)	适配机型	保障架次	效用比例 (%)
YA	528	520	A	12	98.4
YB	240	238	B	6	99.1
YC	0.05	0.04	AB	18	90.1
YD	0.07	0.07	ABC	18	99.6

表 4 设备查询表单

ID	设备类型	设备工作区域	载体	运行状态
0101001	RFID	*号区域*仓库	01-1集装箱	正常
0102009	RFID	*号区域*营房	02-9储存柜	正常
0106003	身份识别器	*号区域*跑道	车牌号*****	移动
0201014	液压传感器	*号区域*油库	*运油储存罐	正常
0205007	温度传感器	*号区域*设施	气象设备	无响应

3.2 平台功能实现

平台功能主要包括经预处理的飞行保障相关数据统计与展示、分析与应用以及平台自身层面的设备管理功能, 具体包括飞行任务计划、保障出动力量、物资消耗数据、时空信息数据的统计与展示、数据统计分析、决策推荐的功能模组和设备载体监控模组[9,10].

态势显示界面是平台的默认初始界面, 即数据平台首页. 通过汇总各数据集中应用频率较高的部分, 将其整合至一个全局概览页面中. 该模块界面如图 4 所示, 主要包括气象数据概览、宏观保障目标、当日放飞架次、保障任务执行情况及问题上报简报等分块以及本场当前保障态势、本场当前保障流程计划、保障力量分布、装备物资消耗及作业情况、保障效能粗统以及基本库存情况统计等分区.

场站飞行保障大数据平台的数据统计与分析模组以数据可视化手段作为直观呈现的主要方法, 在技术层使用 pandas+numpy 进行数据预处理, 使用 ECharts 数据可视化库和 Vue 组件连接数据库实现数据可视

化, 在前端中引入柱状图、折线图、堆栈折线图、数据关联链作为可视化呈现方法, 同时对完成预处理的数据进行选择性抽取, 组成场站飞行保障数据平台前端数据集并建立相关数据列表, 实现末端高需求性采集数据的感知与呈现. 模块共分为 3 部分, 分别由图 5 保障能力通览、图 6 保障效能通览和图 7 外部关联情况构成.

图 4 态势显示界面

图 5 保障能力通览

图 6 保障效能通览

保障能力分析主要依照本场保障需求及上级保障任务规划安排, 以所需具备的各机型作为分类标签, 划分为整体层面的机型保障覆盖能力、机动转场能力和资源再生能力综合评估指标结果、相应的指标上下限与趋势的预测, 以及以机型种类为分类标签, 分别对不同机型机种对应所需的保障分队装备、物资、技术人员、编制数量、作业时长和单场保障架次等保障能力数据进行汇总统计与条件查询; 此外也可以业务类型为区分, 对各业务部门所具备的保障能力、编制装备和物资及相应保障能力与机型适配性进行统计, 最终目的是从能力视角出发, 提供不同维度的场站飞行保障统计数据值细化机型覆盖、资源再生等保障能力综合评估指标, 进一步充实场站飞行保障指挥机关进行保障决策活动的基础数据集.

图 7 外部关联

保障效能分析由保障满足程度分析和保障成本分析两部分构成. 保障满足程度分析的核心依据是飞行计划是否严格实现, 由此确保飞行计划不因保障方面问题导致延误; 或在满足程度未达到目标值的条件下, 查明哪些原因造成了飞行计划未能顺利实施, 其主要分析内容为物资的计划-消耗量及其效用比例、装备的计划-派遣量及其作业时长与需求待时比例等; 保障成本分析主要面向保障资金成本和装备寿命等问题, 通过统计总体物资消耗量与物资效用比例对保障计划中物资的消耗成本进行分析; 通过统计装备及设施的维修维护、保养运输等资金成本及其复用、回收、报废、返厂比例体现场站飞行保障力量建设及运营的实际情况; 结合绩效分析观念设立效益指数、营运指数、负荷指数和发展指数等评价飞行保障效能的相关指标. 通过保障效能分析活动, 能够确保在完成飞行保障计划这一固定收益目标的前提下, 尽可能地辅助指挥层找到影响保障计划完成的关键节点, 从而降低成本, 提高效益, 增强效率, 进一步提高可持续的保障能力.

装备-物资关联分析通过汇总历史统计数据中的机型-装备-物资之间的计划与消耗数据, 利用数据关联规则挖掘技术对隶属不同业务部门的各项物资、装备之间的消耗规律进行知识发现, 选择历史数据中能够直接影响飞行保障计划实施的数据作为主体, 并以机型种类-物资种类及消耗量-装备类型及消耗时长作为三级关联链, 按照关联强度从高到低的顺序提取与该主体消耗或使用高度相关的前 5 类机种、物资及装备, 并按照机型为分类标准提供历史消耗-时长折线图作为参考.

3.3 平台性能分析

大数据平台的性能分析主要内容及评判标准包括: 基本功能实现情况、数据导入导出能力、用户交互等级、容错性和可扩展性、线性的计算能力、计算资源分配能力等方面. 考虑到目前场站飞行保障大数据集群环境和大数据平台尚处在研发阶段, 并未实施部署, 故选取当前较流行的基于 MapReduce 计算引擎的大数据平台和基于 Flink 引擎的大数据平台与本文数据平台性能进行对比分析, 并结合场站工作实际情况, 说明选取 Spark 作为构建飞行保障数据可视化平台的计算引擎的原因.

本节的性能分析测试使用 BDEv 控制数据集 [11], 具体配置情况如表 5 所示, 采用 kafka-producer-perf 测试平台吞吐量-负载能力并作为性能评价指标; 之后通过对比数据处理时间, 对 MapReduce、Spark 和 Flink 的数据处理能力进行比较.

Throughput-Load_users 测试中, 将执行参数设置为: replication-factor=2; acks=1, 以批处理大小 batch_size 作为变量表示平台负载能力, 在无数据压缩条件下测试各平台吞吐量, 所得结果如图 8 所示.

对图 8 结果进行分析, 其中 α_1、α_2、α_3 分别代表折线的上升、平稳和下降 3 个区域. α_1 区域面积越大, 说明系统的性能能力越强; α_2 面积越大, 说明系统稳定性越好; α_3 面积越大, 说明系统的容错能力更强. 因此总体上来看, MapReduce 引擎的性能相较于其他二者具有明显差距.

考虑到本文数据处理流程的需求, 故选择以 WordCount 和 Grep 为基准 [12] 对系统 CPU 做限制. 图 9、图 10 分别对应了 3 个引擎在 Grep 和 WordCount 基准任务中, 不同虚拟节点数量条件下处理数据集消耗的时长. 从图中可以看出, MapReduce 引擎的绝对性能相较于其他二者具有明显差距, 在两项任务中均体现出较弱的处理性能. 以上结果表明, 从系统整体资源性能来看, MapReduce 虽然技术成熟, 但与 Spark、Flink 相比, 在经济性、时效性等方面已逐渐无法满足大数据对平台基础能力的要求. 接下来从三者深层结构层面进行分析.

表 5 测试框架配置具体情况

框架	参数	参数值
MapReduce	DB_size	128 MB
	DR_factor	2
	Heap_size	2.0 GB
	Mapper	2
	Reducer	2
	.parallellcopies	10
	IO_sort	512 MB
	IO_sort_overflow	80%
Spark	DB_size	128 MB
	DR_factor	2
	Heap_size	7.8 GB
	Worker	1
	Corer	4
Flink	DB_size	128 MB
	DR_factor	2
	Heap_size	7.8 GB
	TM_num	4
	TM_core	4
	NetworkBuffer	512
	Predistribution	—
	IO_sort_overflow	80%

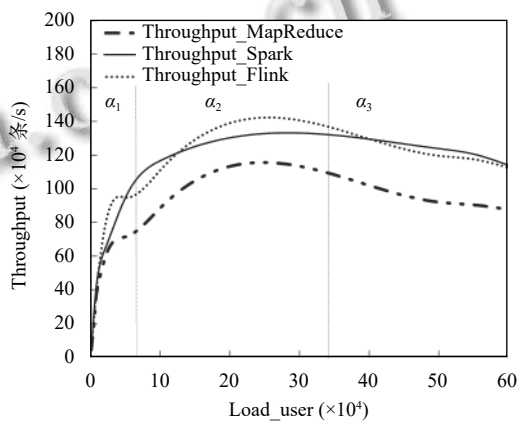

图 8 Throughput-batch_size 变化情况

图 9 表明在 Grep 中, Spark 和 Flink 的性能大幅大幅领先 MapReduce. 最重要的原因是 MapReduce 的 API 在该基准中使用两个 MapReduce 进程并分别用于搜索和排序, 内存资源占用较大. Spark 和 Flink 则都能够通过过滤函数匹配输入项, 之后通过行计数方式借助完成内存排序, 避免了重复复制流程. 此外, map( )

函数执行模式匹配时仅能占用分配内存的 50%, 而 Spark 和 Flink 的并行度被设为集群中的 CPU 核心的总数量, 可用计算资源也远超过 MapReduce.

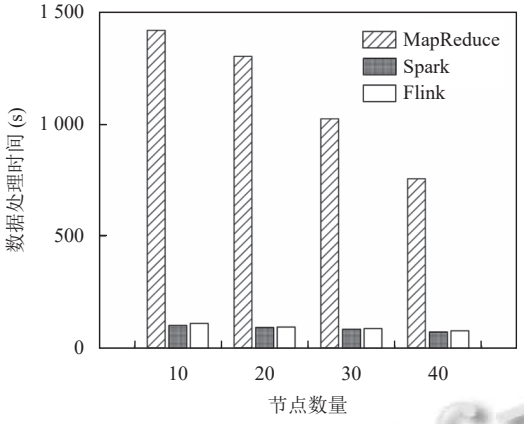

图 9 Grep 下平台性能与数据处理能力比较

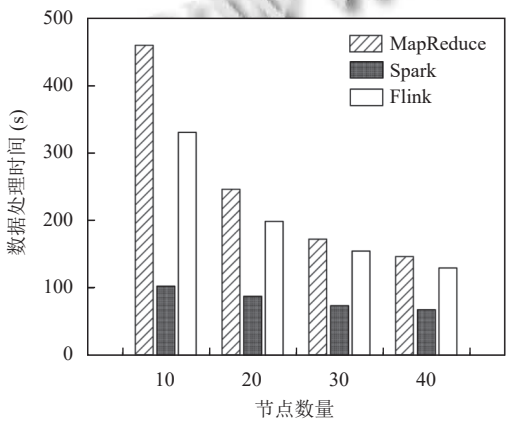

图 10 WordCount 下平台性能与数据处理能力比较

图 10 中由于 Spark 在 WordCount 中由其 API 提供的 reduceByKey() 函数能够对每个词语的出现次数进行加和. 而 Flink 则使用 groupBy().sum() 方法, 针对此类工作负载的优化较差, 与其他基准相比 WordCount 对 CPU 的约束性降低了 Flink 的内存优化效果, 带来了额外的计算开销, 导致 Flink 在该任务中的性能滞后.

平台节点扩展能力方面, Spark 通用的扩展方式与初始配置方法基本一致, 本文通过克隆子节点虚拟机的方式完成新增用户、修改节点 host 及 IP 配置、配置免密登录及环境变量配置活动, 并在配置文件中相关的新增节点并格式化. 现在原基础上保持表 5 中 Spark 引擎配置环境不变, 新增一个子节点, 总结对比集群 Size、计算资源占用大小、文件系统可利用大小及占用率 4 个方面如表 6 所示. 之后执行 start-balancer.sh 进行扩充节点后的负载均衡并进行验证, 主要对比内容为文件系统使用量对比、DFS 未占用量对比、Blocks 块存储量及池占用量, 验证结果统计在表 7 中.

表 6 Filesystem 单节点扩容前后情况对比

Expand	Size (GB)	Used (GB)	Available (GB)	Use (%)
F	7.85	5.46	1.46	69.5
T	11.7	5.75	4.40	49.2

表 7 单节点扩容后负载均衡结果对比

Slave	Expand	Used (GB)	Non DFS used	Blocks	Blocks pool used (%)
1	F	3.281	0.642	41	83.5
	T	**3.183**	**0.74**	**38**	**81.1**
2	F	3.274	0.649	37	83.42
	T	**3.183**	**0.74**	**37**	**81.1**
3	F	2.996	0.926	35	76.36
	T	**3.183**	**0.739**	**38**	**79.8**

最后通过增加节点的数量增大并行计算处理器计算负载, 对扩容后平台进行负载均衡处理, 同时控制数据规模保持不变, 使用 Amdahl 定律对平台并行计算加速比进行分析. 以 slave=2 为基准, 统计各情况下平台主要子任务的时间占用比例, 并将结果统计如图 11 所示, 对该框架运行环境下平台扩容效能及优化重点进行评估.

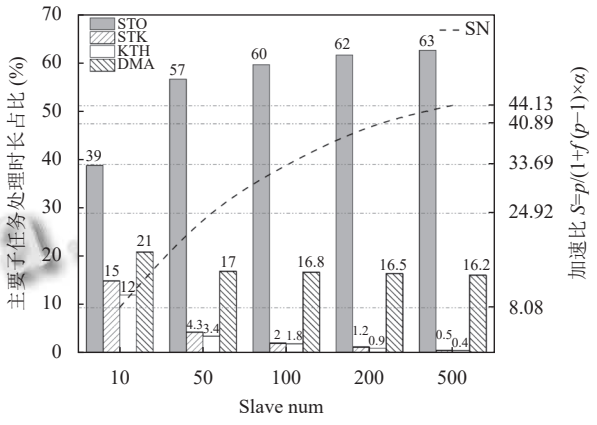

图 11 处理器-加速比-子任务时长占比情况

图 11 中统计的子任务分别为: SQL to Oracle (STO)、SQL to Kafka (STK)、Kafka to Hive (KTH) 以及 Data Mining Algorithm (DMA) 四项时长占比较大的任务, 从图中统计的节点-加速比及节点-时长占比统计情况可以得出以下结论:

(1) 加速比理论应满足 $\lim_{n\to\infty} S = \frac{1}{f}$, 但因并行计算时的通信、同步和归约的操作所花费的额外开销时间使得加速比上限从统计结果来看存在上界. 由于测试集

所需的计算资源有限, 在虚拟节点数量增加的初始阶段加速比提升性能明显.

(2) 主要子任务中, STO 属于串行计算任务, 其余 3 项属于可并行计算任务, 且 DMA 任务主要依托于规模性的 GPU 计算. 因此随着虚拟节点数量的增加, STK 与 KTH 任务的执行效率得到了明显的优化, 但对 DMA 任务带来的 CPU 优化效果有限. 而 STO 作为串行计算任务, 虚拟节点数量的增加并不会对其运算性能产生实质性的影响, 故处理时长基本不变.

(3) 将 (2) 中结果归纳为主要子任务的处理时长占比, 4 大子任务从 10 节点数量时的 87% 占比, 到 500 节点数量时已经下降到 80.1%, 由此得出其余任务同样以串行计算为主, 不需要着重考虑扩容优化问题; 随着总体处理时长的下降, 串行计算任务的时长占比明显提升, 纯并行计算任务 STK、KTH 由于优化效果存在下界, 当节点数量大于 100 后继续选择扩容方法进行优化的效益逐渐降低, 仅有 DMA 任务还具备一定的优化潜力. 故当本实验环境下虚拟节点数量低于 100 时, 选择扩充虚拟子节点能够对平台数据处理效能有较大的提升, 并具有较高的效益.

综合以上分析结果, 本文选择 Spark 计算引擎, 基于 Hadoop-Yarn 资源管理架构构建场站飞行保障数据可视化平台的设计理念行之有效, 并且具备一定范围内的扩容潜力.

4 结论与展望

针对当前航空兵场站信息化建设中的数据应用南、挖掘手段单一的现状, 综合分析当前保障数据处理与应用需求, 引入大数据技术, 提出了基于 Spark 计算引擎的场站飞行保障数据可视化平台设计方案. 该平台在场站局域网集群环境下, 利用 Flume 和 Sqoop 完成末端保障数据和信息系统数据的采集任务, 利用 Kafka 完成数据采集设备和载体的状态数据监控与传输, 并通过前端框架 Vue 中嵌入 ECharts 图表库的方式完成数据可视化呈现, 使得场站飞行保障活动数据的应用相比传统基于人工表格汇总的信息孤岛情况, 在工作效率和数据获取与管理呈现能力方面有了巨大的改善和提升.

由于各方面因素限制, 平台仅支持批数据的处理与分析, 在数据获取与管理方面需要进行多次且长期的上传行为, 在使用效率上仍有待改进, 同时采集数据中包含的时间戳与空间坐标数据, 也为平台提供了向时间序列与实时处理模式改进的可扩展性. 除此之外, 平台应用场景的特殊性也要求其数据流程必须更进一步关注数据的安全性和保密性, 这也是未来数据平台在场站实际应用所必须具备的基础要求.

参考文献

1 毛允杰, 李云果, 迟蒙. 浅析现代联合作战组织指挥决策大数据库的建设. 海军学术研究, 2018, 345(3): 21–22.

2 王洪威, 张元杰. 大数据背景下战区战场环境信息保障体系构建与运用探讨. 作战测绘保障, 2018, (4): 13–14.

3 肖治鑫, 杨西龙, 姜玉宏. 基于大数据技术的战备物资储备量建模与分析. 舰船电子工程, 2019, 39(10): 132–137. [doi: 10.3969/j.issn.1672-9730.2019.10.030]

4 于雨. 大数据在空军场站飞行保障中的应用研究 [硕士学位论文]. 徐州: 空军勤务学院, 2019.

5 Mior MJ, Salem K. ReSpark: Automatic caching for iterative applications in Apache Spark. Proceedings of the 2020 IEEE International Conference on Big Data (Big Data). Atlanta: IEEE, 2020. 331–340.

6 张飞. 一种基于 Kafka 的数据采集与实时处理系统的设计与实现 [硕士学位论文]. 西安: 西安电子科技大学, 2019.

7 朱二华. 基于 Vue.js 的 Web 前端应用研究. 科技与创新, 2017, (20): 119–121.

8 陈卓然. 基于 Hadoop 的票务分析系统设计与实现 [硕士学位论文]. 北京: 北京邮电大学, 2021. [doi: 10.26969/d.cnki.gbydu.2021.000542]

9 刘君阳. 多机型飞行保障流程优化问题研究. 电脑与信息技术, 2022, 30(3): 30–32, 36. [doi: 10.19414/j.cnki.1005-1228.2022.03.019]

10 朱世松, 刘君阳. 基于 DBN 的飞行保障力量配置辅助决策方法. 系统仿真学报, 2022: 1–11. [doi: 10.16182/j.issn1004731x.joss.22-0037]

11 代明竹, 高嵩峰. 基于 Hadoop、Spark 及 Flink 大规模数据分析的性能评价. 中国电子科学研究院学报, 2018, 13(2): 149–155. [doi: 10.3969/j.issn.1673-5692.2018.02.007]

12 姜春宇, 魏凯. 大数据平台的基础能力和性能测试. 大数据, 2017, 3(4): 37–45.

(校对责编: 牛欣悦)