

DataX 工具在新冠肺炎数据上报中的应用

田翠姣, 苏义武

(武汉大学中南医院, 湖北 武汉 430071)

摘要:新冠肺炎疫情防控期间,精确的数据上报越发显得重要,国家医政医管局发文要求每天定时上传确诊和疑似患者全量医疗数据。该文重点讨论了利用 DataX 工具进行数据提取的方法以及该工具是否具有推广价值。搭建 DataX 运行环境,编写测试脚本并运行,检测输出数据各项指标,观测任务执行时间,并与之前手工模式进行对比。使用基于 DataX 的工具环境后,实际执行效率明显高于原先手工执行 SQL 语句导出数据的方式。DataX 不仅能快速导出数据,而且能将不同数据库的数据抽取到目标库,实现数据的整合,此外,数据导出高效、准确,除了较好地完成本次上报任务外,还能满足医疗机构日常工作中各类数据上报的需求。

关键词:新冠肺炎;DataX;CSV;JSON;数据上报

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2020)11-0216-05

doi:10.3969/j.issn.1673-629X.2020.11.040

Application of DataX Tool in Data Reporting of New Coronavirus Pneumonia

TIAN Cui-jiao, SU Yi-wu

(Zhongnan Hospital of Wuhan University, Wuhan 430071, China)

Abstract: During the prevention and control of novel coronavirus pneumonia, accurate data reporting is becoming more and more important. The National Medical Administration and Hospital Authority issued a document requiring that full medical data of confirmed and suspected patients be uploaded regularly every day. We mainly discuss the method of data extraction using DataX tool and whether the tool is promoted. Set up DataX running environment, write test script and run it, test output data indicators, observe task execution time, and compare with previous manual mode. After using the tool environment based on DataX, the actual execution efficiency is significantly higher than the original way of manually executing SQL statements to export data. DataX can not only export data quickly, but also extract data from different databases to the target database to realize data integration. In addition, data export is efficient and accurate. Besides completing the reporting task, it can also meet the needs of various data reporting in the daily work of medical institutions.

Key words: new coronavirus pneumonia; DataX; CSV; JSON; data reporting

1 背景

2020年2月15日国务院医疗救治组发文《关于做好武汉市新冠肺炎患者医疗数据报送工作的通知》,通知要求武汉市全部新冠肺炎定点医院自2月15日起每天中午12点前上报新冠肺炎在院病例和出院病例的全量医疗数据。数据包括但不限于电子病历数据、病程记录数据、护理记录数据、检验信息数据、医嘱信息数据、重症系统数据、影像报告系统数据、手术麻醉系统数据、病案首页数据。

数据上报采用VPN+FTP方式,数据文件要求每

张表数据存放到独立的CSV格式文件,并且文件首行为数据表对应字段名,每行的各个字段之间使用Tab符(\t)分隔,数据字符集要求UTF-8,最后文件用gz压缩为单个压缩文件^[1]。

2 方案分析

按照要求对任务进行了分析,上传数据文件要求为CSV格式文件。CSV格式文件是一种通用的、相对简单的文件格式,在商业和科学中应用广泛,其特点是分隔的数据格式,字段之间的分割格式有逗号分隔和

收稿日期:2020-03-05

修回日期:2020-07-05

基金项目:湖北省科技惠民计划项目(2017ACB640)

作者简介:田翠姣(1968-),女,硕士,副主任护师,研究方向为数据统计、大学生教育、护理心理学;苏义武(1986-),男,工程师,研究方向为医院信息化。

空格分隔两种^[2-3]。数据内容方面主要涉及到 9 个信息系统,4 种不同的数据库,进一步分解任务后得出需要上报的表格数量为 22 个(见表 1)。

表 1 上报数据信息数据库格式

要求上报数据	系统数据库格式
电子病历数据	Oracle
病程记录数据	Oracle
护理记录数据	MySQL
检验信息数据	Oracle
医嘱信息	Oracle
重症系统数据	MongoDB
影像报告系统数据	SQL Server
手术麻醉系统数据	SQL Server
病案首页数据	Oracle

按照以往提取数据的方式,决定先用最熟悉的 Oracle 数据库做测试,使用 Oracle 自带的 PL/SQL 工具查询出数据后,直接导出 CSV 文件。经过测试发现这些 CSV 文件默认使用逗号分隔符,字符集为 GB2312,需要使用特定工具进行转换。此种方法不仅繁琐,费时费力,而且会出现转换失败的情况。经过不断摸索,最后选择使用 DataX 工具进行数据批量提取的方法。

3 相关工具及安装介绍

DataX 是阿里巴巴集团内被广泛使用的离线数据同步工具,能够实现包括 MySQL、Oracle、SQL Server、MongoDB、HDFS、Hive、OceanBase、HBase、OTS、ODPS 等各种异构数据源之间高效的数据同步功能^[4-6]。DataX 将复杂的网状的同步链路变成了星型数据链路,DataX 作为中间传输载体负责连接各种数据源。当需要接入一个新的数据源的时候,只需要将此数据源对接到 DataX,便能跟已有的数据源做到无缝数据同步。更重要的是下载即可用,配置简单,支持 Linux 和 Windows,只需要短短几步就可以完成数据的传输^[7]。

另外需要引入的工具是 Python,Python 是一种跨平台的计算机程序设计语言,是一种面向对象的动态类型语言,最初被设计用于编写自动化脚本(shell),随着版本的不断更新和语言新功能的添加,越来越多地被用于独立的、大型项目的开发^[8]。再加上 Java 运行环境,组成了 DataX + Python + Java 环境,该文以 Windows10+Datax3.0+Python2.7+jdk1.8 组合环境进行说明。

先安装 jdk 环境并配置系统环境变量,然后缺省安装 Python 软件并配置系统环境变量,最后解压

Datax 压缩包到指定目录(此处路径为 D:\datax),至此环境配置完成,整个安装过程为纯傻瓜式安装,没有特殊设置,简单易行。

4 DataX 运行原理

在 DataX 框架中,数据交换通过 DataX 进行中转,任何数据源只要和 DataX 连接上即可以和已实现的任意数据源同步。

DataX 作为离线数据同步框架,采用 Framework+plugins 架构构建。将数据同步过程中的读取和写入过程抽象为内部 Reader/Writer 插件,如图 1 所示。

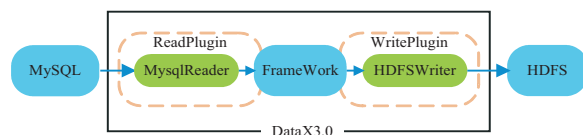


图 1 DataX 框架结构

Reader:数据采集模块的抽象,负责采集数据源数据并发送给 Framework。Writer:数据写入模块,不断从 Framework 取数据并写入目标端。Framework:作为 Reader 和 Writer 的数据传输通道,处理缓冲流控、并发、数据转换等核心技术问题。

框架提供了简单的接口与插件接入机制,只需要任意加上一种插件,就可以无缝对接其他数据源。开放式的框架也让开发者可以在短时间内开发一个新的插件,以快速支持新的数据源同步,以下结合此次数据上报应用具体说明。

此外,DataX 内部运行采用 Job 模式运行机制(如图 2 所示),DataX 完成单个数据同步的作业,称之为 Job,DataX 接受到一个 Job 之后,将启动一个进程来完成整个作业同步过程。DataX Job 模块是单个作业的中枢管理节点,承担了数据清理、子任务切分(将单一作业计算转化为多个子 Task)、TaskGroup 管理等功能。

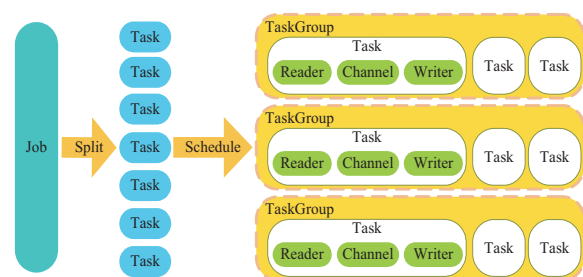


图 2 DataX 各功能模块关系

DataX Job 启动后,会根据不同的源端切分策略,将 Job 切分成多个小的 Task(子任务),以便于并发执行。Task 便是 DataX 作业的最小单元,每一个 Task 都会负责一部分数据的同步工作。

切分多个 Task 之后, DataX Job 会调用 Scheduler 模块, 根据配置的并发数据量, 将拆分成的 Task 重新组合, 组装成 TaskGroup(任务组)。每一个 TaskGroup 负责以一定的并发运行完毕分配好的所有 Task, 默认单个任务组的并发数量为 5。

每一个 Task 都由 TaskGroup 负责启动, Task 启动后, 会固定启动 Reader→Channel→Writer 的线程来完成任务同步工作。

DataX 作业运行起来之后, Job 监控并等待多个 TaskGroup 模块任务完成, 等待所有 TaskGroup 任务完成后 Job 成功退出。否则, 异常退出, 进程退出值非 0。

5 脚本编写及运行

DataX 使用一个 JSON 文件来描述一个 Job^[9-10], 通过配置 JSON 格式的文件, 使用 Python 命令就可以启动 DataX 执行任务。这个 Job 本身配置并不复杂, Reader 插件和 Writer 插件基本就是配置连接数据库使用的连接串、用户名、密码、查询表、字段名。接着, 按照 JSON 编码规则^[11-12], 编写如下脚本, 文件名为 demo.json, 存储目录为 D:\datax\

```
{
  "job": {
    "content": [
      {
        "reader": {
          "name": "mysqlreader",
          "parameter": {
            //数据库用户名
            "username": "db_user",
            //数据库密码
            "password": "db_pass",
            //以下为查询字段,若查询全部字段,可以使用*代表字段列表"
            "column": [ "patientno", "deptcode", "name", "rptext" ],
            "connection": [
              {
                //数据源数据表名称
                "table": [ "pacs_report" ],
                //数据库连接串",更多串配置参考官网示例
                //MySQL 数据库连接串
                //"jdbcUrl": [ "jdbc:mysql://IP 地址:3306/database" ]
                //Oracle 数据库连接串
                //"jdbc:oracle:thin:@ IP 地址:1521/实例名",
                //SQLServer 数据库连接串
                "jdbc:sqlserver://IP 地址:3433;DatabaseName=dbname"
              }
            ]
          }
        }
      }
    ]
  }
}
```

```
"writer": {
  "name": "txtfilewriter",
  "parameter": {
    //导出数据文件输出路径
    "path": "D:\\datax\\myoutput ",
    //导出数据文件名称
    "fileName": "pacs_report",
    //数据内容字符集
    "encoding": "UTF-8",
    //覆盖之前同名文件模式
    "writeMode": "truncate",
    //日期格式
    "dateFormat": "yyyy-MM-dd HH:mm:ss",
    //导出数据文件格式
    "fileFormat": "csv",
    // csv 分隔符为\t,即 tab 方式分割
    "fieldDelimiter": "\t",
    //压缩格式 gzip,能减少生成文件的大小
    "compress": "gzip",
    //表头设置
    "header": [ "patientno", "deptcode", "name", "rptext" ],
  }
}
```

脚本运行也非常简单,在 CMD 窗口模式下,输入如下语句执行:python D:\datax\bin\datax.py D:\datax\pacs_report.json,其中 datax.py 为 python 安装后自带的解析脚本。整个命令大意为使用 python 命令利用 datax.py 脚本执行 pacs_report.json 配置的任务^[13-14]。此时在 JSON 脚本中设置的目录 D:\datax\myoutput 下会出现 pacs_report.csv.gz 压缩文件,压缩文件内为 pacs_report.csv 文件。其余的 21 个文件,按照相同的方式配置 job,最后将所有运行脚本形成批处理文件,一键执行后生成所需 22 个数据文件。

6 效果对比

经过自己检测,数据内容、文件格式、分隔符、字符集、压缩形式均符合上报要求,上报后经国家医政医管局工程师检验,亦没有问题。效率方面,原先手工生成方式,每次都需要执行大量的 SQL 语句,并且需要转换、压缩、校验,一个文件生成至少需要 2 分钟,并且是在数据量较小的前提下,费时费力而且容易出错。使用基于 DataX 的工具环境后,实际执行 33 万条费用明细数据全表导出,只需要 27 秒即完成,全部 22 个数据文件生成平均花费 3 分钟,极大提高了数据收集的效率。

7 扩展应用讨论

7.1 疫情数据上报

随着疫情发展,疫情数据报送内容、对象、形式不断变化,高峰时期对外报送报表数目多达 20 余个。如每日 8 点邮件报送住院人数、发热门诊人数,每日 10

点网络报送核酸检测数量、中医药治疗病例数,每日 18 点邮件报送住院病人明细等等(部分任务清单如表 2 所示)。报送对象涉及国家、省、市、区的卫计委相关部门或者各级防控指挥部。

表 2 疫情数据上报任务清单

时间	工作内容	上报对象	上报方式
10 时	新冠核算检测信息日报表—省综合统计信息平台	疾控中心、卫健委	QQ 报送、网络报送
12 时前	电子病历记录、费用记录、医嘱记录、手麻记录、LIS、PACS、重症系统等全量医疗数据	国家医政科	VPN+FTP 方式
18 时前	《0-24 时新型冠状病毒感染的肺炎确诊病例信息日报表》	武昌区医政科	邮箱报送
18 时前	《武昌区医疗机构每日病例信息表》	武昌区医政科	邮箱报送
19 时前	发热门诊日报表-疫情直报系统网上报	武汉市卫健委	网络报送
19 时前	住院病情明细表-疫情直报系统网上报	武汉市卫健委	网络报送
19 时前	住院床位日报表-疫情直报系统网上报	武汉市卫健委	网络报送
19 时	存量疑似病例进行排查		网络报送
19 时	市级定时医院床位数统计表	武汉市卫健委	
24 时	新冠肺炎患者评估一览表	武汉市卫健委	邮箱报送
甲类 2h 内	传染病报告卡,见《传染病信息报告管理规范(2015 年版)》	疾控中心	网络填报
甲类 2h 内	尸体消毒、转运报告表	江夏区民政局、疾控	邮箱报送
...

从表 2 任务清单可知,疫情数据上报大多采用网页填报或邮箱报送方式,并且对实时性要求高。由于并未提供数据接口,初期采用人工方式从各信息系统将数据整理出来,十分繁琐、耗时。而且涉及到的系统比较多,数据库类型也各异,为此,利用 DataX 工具强大的离线同步功能将所需数据汇总到一个专用数据库(此处使用 Oracle)中,然后引入网页数据填写成熟技术或者 Python 工具,梳理出网页填报数据与汇总数据库中信息数据对应关系。进行数据统计上报时,自动提取汇总数据库中相关数据,填充到卫生部门下发的各类疫情报表。最后系统启动代码或工具将数据填写到网页上的相应元素域或邮箱中,数据填写完毕后,自动提交并退出网页^[15-16]。通过该方案实现了大部分数据的自动化填报,极大减少了数据上报的工作量。

7.2 疫情数据展示

精准及时的数据有利于疫情处理。此次疫情对医疗资源消耗巨大。为了提高床位利用效率、缓解医疗资源紧张的形势,根据医院差异化定位分级接诊、促使患者快速分流势在必行,这离不开全面数据分析的支持。

为了能够让院领导实时获取所有院区各项关键指标,便于快速精准决策,工程师利用大数据分析手段开发了移动 BI 平台,用于展示疫情专题数据(如图 3 所示)。并在此基础上开发出院隔离患者初筛报表,按



图 3 方舱医院疫情专题主界面

照诊疗标准中的出院条件自动化标识拟离院患者,然后将报表实时推送给相关领导,使得管理者对全院心中有数、决策有据,为改善出院管理、床位调配、转院交接等方面发挥了作用。重视效率的同时,质量和安全也毫不松懈,针对住院日、吸氧、ICU 入住、转归等情况开展实时统计,如若发现负性指标及时通报同时追查。

BI 平台需要实时抽取各类系统大量的数据,这涉及到多个无法回避的问题。第一,一些数据,比如患者重症情况、吸氧情况、CT 检查结果、核酸检查结果分布在不同的数据库中,无法通过联合 SQL 查询的方法及时获取数据;第二,由于数据的及时性要求非常高,如果采取直接从生产数据库中抽取数据的方式,势必会造成生产库压力骤增,进而影响全院信息系统的运行效率,从而对医疗业务产生影响。

DataX 工具在此时发挥了巨大的作用,按照数据分型编写好数据抽取脚本后,根据数据产生规律定制化 Job 计划任务。数据实时同步到 BI 数据仓库,BI 前端展示只需要查询数据仓库即可快速查询数据,而且由于是高效率地实施同步模式,数据延迟在可控范围内。

7.3 其他应用

疫情结束后,数据上报会停止,如果仅仅为了解决这一次上报问题,花费精力去配置工具有些得不偿失,但是发现在医院实际工作中经常有各种数据上报,上报数据格式大多数都是 CSV 格式。比如国家卫生统计 4-1 报表上报病案首页数据, HQMS 医疗数据上报、全国三级公立医院绩效考核医疗数据上报、流感数据上报等都涉及到 CSV 格式数据报送,特别是有一些数据还涉及到不同数据库表联合查询,普通的数据库查询导出方式很难实现,且容易出现数据丢失、错位的问题。实践表明,DataX 不仅能快速导出数据,而且能将不同数据库的数据抽取到目标库,形成数据整合,数据导出高效、准确,能很好地满足医疗机构日常工作中各类数据上报需求。

8 结束语

DataX 作为一个服务于大数据的 ETL 工具,除了提供数据快照搬迁功能之外,还提供了丰富数据转换的功能,让数据在传输过程中可以轻松完成数据脱敏、补全、过滤等数据转换功能,另外还提供了函数,让用户自定义转换函数。结合 Python 和 JDK 运行环境^[17],不仅部署简单,而且操作方便,能够根据需要整合各类格式数据,甚至是 txt 文本、ftp 文件,对于医学统计专业常用的 dbf 格式文件,也能突破常规 256 列

数据的限制,对于医疗大数据分析意义重大^[18]。在此次新型冠状病毒引起的肺炎疫情中,本院也将该方案推荐给其他医疗机构,及时、高效地完成了数据上报工作,为消灭疫情提供了坚实可靠的数据支撑。

参考文献:

- [1] 陈衍鹏. 基于 Python 第三方库实现 Excel 读写[J]. 微型电脑应用, 2017, 33(8): 75-78.
- [2] 丁亚涛. 基于 CSV 格式的考试系统研究[J]. 电脑知识与技术, 2015, 11(28): 70-71.
- [3] IT_小马哥. 简书. CSV 文件[EB/OL]. 2019-05-20[2020-03-02]. <https://www.jianshu.com/p/7d15ff418310>.
- [4] Alibaba. GitHub. alibaba/DataX[EB/OL]. 2019-12-02[2020-03-02]. <https://github.com/alibaba/DataX>.
- [5] MongoDB Inc. The MongoDB 3.0 manual[EB/OL]. 2015. <https://docs.mongodb.org/manual/>.
- [6] WARREN SLESINGER. ORACLE[J]. The Yale Review, 2019, 107(3): 42.
- [7] 青哥 DevOps. 简书. DataX 使用指南[EB/OL]. 2018-12-04[2020-03-02]. <https://www.jianshu.com/p/ecbfe77d33ee>.
- [8] 杨凯利, 山美娟. 基于 Python 的数据可视化[J]. 现代信息技术, 2019, 3(5): 30-31.
- [9] UCEDA-SOSA R. Java XML and JSON[J]. Computing Reviews, 2017, 58(2): 71-72.
- [10] BURG G J J, NAZÁBAL A, SUTTON C. Wrangling messy CSV files by detecting row and type patterns[J]. Data Mining and Knowledge Discovery, 2019, 33(6): 1799-1820.
- [11] 梁志宏, 陆歌皓, 黄宇翔, 等. 基于 JSON 的异构数据集成的研究[J]. 计算机科学与应用, 2018, 8(3): 314-322.
- [12] 穆鑫鑫, 蒋同海, 程力, 等. 基于 JSON 的离线数据同步策略及应用[J]. 计算机系统应用, 2017, 26(12): 257-261.
- [13] 聂晶. Python 在大数据挖掘和分析中的应用优势[J]. 广西民族大学学报: 自然科学版, 2018, 24(1): 76-79.
- [14] 李强, 白建荣, 李振林, 等. 基于 Python 的数据批处理技术探讨及实现[J]. 地理空间信息, 2015, 13(2): 54-56.
- [15] 长江云. 中南医院信息中心统计室“组合拳”援前线稳大局[EB/OL]. 2020-04-12[2020-04-16]. <http://m.hbvt.com.cn/p/1827547.html>.
- [16] weixin_38784605, CSDN 博客. RPA 工具实现网页内自动填报和上传资料[EB/OL]. 2019-09-18[2020-04-16]. https://blog.csdn.net/weixin_38784605/article/details/100989722.
- [17] Oracle, Java Documentation - Get Started[EB/OL]. 2018. <https://docs.oracle.com/en/java/index.html>.
- [18] 关金金, 未培, 庄彦. 基于 Hadoop 的海量数据处理平台的架构与研究[J]. 科技视界, 2019(20): 99-100.