

黄志,李涛,宋瑶,等.基于 Json 的小型异构数据库同步策略研究[J].气象研究与应用,2020,41(1):48-53.
Huang Zhi, Li Tao, Song Yao, et al. Research on synchronization strategy of small heterogeneous database based on Json [J].Journal of Meteorological Research and Application,2020,41(1):48-53.

基于 Json 的小型异构数据库同步策略研究

黄 志,李 涛,宋 瑶,苏传程
(广西区气象信息中心, 南宁 530022)

摘要: 针对不同气象业务系统所辖台站参数表在表字段不同的情况下,如何快速完成各个台站参数表之间的数据同步变更问题,本文提出了一种小型异构数据库同步方案。通过将数据库表名、表字段以及数据库连接信息封装为 Json 格式键值对配置文件,采用 SQL 列别名的方式完成对数据库表的查询;当元数据表参数信息发生变动时,通过对 Json 键值对顺序分解匹配,能准确定位元数据表和目标表各自对应的表字段,绑定其被修改的参数值,并自动生成 SQL 语句完成增删改操作。结果表明,基于 Json 的台站参数数据同步方式可简化开发框架并有效减低开发成本,快速建立各个异构数据库表的台站参数数据同步流程,通过在系统外部配置 Json 文件的方式可以降低后台代码的耦合度,有利于后期系统的管理和运维,具有良好的推广应用价值。

关键词: 台站参数;同步;Json;列别名

中图分类号: P49 **文献标识码:** A **doi:** 10.19849/j.cnki.CN45-1356/P.2020.1.10

引言

经过多年的气象信息化建设,各单位新增了不少重要业务系统,在系统建设初期,建设单位都把重点放在如何发挥系统效益上,由于顶层设计缺少前瞻性,对共用的台站参数元数据表没有采用规范的管理设计方式,在数据库表字段设计上缺乏严谨性,导致出现了大量异构的数据库系统和台站参数信息表。各类新增建设台站随着气象信息化发展不断增多,伴随台站搬迁、台站废除等事件引发的台站参数信息变更时有发生,目前气象部门缺少能灵活管理各个业务系统所辖台站参数与台站参数元数据同步管理机制。以往采用的手工编写 SQL 语句完成数据同步方式很难及时准确地做到台站参数的同步更新,给各类气象台站参数数据管理带来不便,台站参数数据同步滞后会导致如经纬度数据有出入造成地理信息显示的错误,在遇到重大天气过程出现时,导致决策服务材料不能为政府应急部门提供精准定位,有可能造成人民生命财产的重大损失。如何将

台站参数变更信息快速准确同步到相关业务系统中是一项重要和迫切的工作。

目前国内外几家大型的数据库厂商提出的异构数据库同步方案,如 Oracle 的透明网关技术、IBM 的 CCD 表、微软公司的出版者/订阅者方案等。国内的如中关村科技软件有限公司、华中科技大学的 Panorama 系统都有过类似系统的研究。上述方案致力于解决异构数据库间复杂的交互操作,但是流程复杂技术难度大而全,并占用庞大的软硬件资源和需要高昂的维护费用,并不符合一些中小企业只针对系统在数据层的小部分数据的同步处理需求。国内学者如穆鑫鑫^[1]采用 Json 格式设计指定数据交换协议并完成轻量级离线数据的同步,取得较 XML 格式更高的数据同步效率;一些国内的银行使用 Json 格式设计数据交换模型支付系统,并取得良好的使用效果,具有效率更高占用内存更小的优点^[2]。研究表明,Json 格式简洁明了结构简单,在轻量级数据管理和传输过程中具有较好的效率优势。因此,本文提出了一种基于 Json 的数据同步策略,构建高效简

收稿日期: 2020-01-18
基金项目: 完善省级智能网格预报业务系统数据环境(桂气科 2019M10)、面向互联网应用的 cimiss 数据同步技术研究(2017M05)
作者简介: 黄志(1981-),男,硕士研究生,高级工程师,研究方向:气象信息化与大数据研究与应用。

捷的数据同步流程,有效降低开发难度,从而实现台站元数据参数信息表与目标库台站参数信息表的数据同步更新。

1 Json 数据格式的优点

Json(JavaScript object notation)是一种轻量级简单的数据交换格式,可以直接使用 JavaScript 来解析,采用完全独立于编程语言的文本格式来存储和表示数据。与 XML 等类似文件格式相比,Json 具有更好的简单性和灵活性,非常易于阅读和编写,同时也易于机器的解析和生成,因为 Json 使用了数据分隔符与 JavaScript 引擎对数据结构(如字符串、数组、对象)的内部表示恰好与这些符号相同,有效简化了数据访问难度。Json 的另一个优点是它的非冗长性,在 XML 中,打开和关闭标记是必需的,这样才能满足标记的依从性;而在 Json 中,这些要求只需通过一个简单的括号即可满足。在包含有大量字段的数据交换中,传统的 XML 标记将会延长数据交换时间,而通过简单的字节数比较发现,对于等效的 Json 和 XML 有效负载,前者总是小于后者,因而 Json 比 XML 具有更高的线上传输效率。

2 基于 Json 的台站参数同步流程研究与设计

针对目前多个气象业务系统中存在多样化的异构台站参数信息表的情况,首先应构建一个数据准确无误的台站元数据参数信息表作为校对依据,当台站元数据某类参数的数据发生变更的时候,能快速完成台站元数据参数与目标台站参数(被修改)数据两者之间的数据库引擎匹配连接,并且准确定位台站元数据参数和目标台站参数各自对应的数据库表字段,绑定其被修改的参数值,然后自动生成匹配的 SQL 操作语句完成对台站元数据参数与目标台站参数的修改。通过修改外部配置文件即可抵消因数据库架构变动和数据表库结构的变更对软件系统代码造成的影响,上述内容是本文需要研究和解决的关键问题。

2.1 台站元数据参数信息表设计

按自定义标准建立台站元数据参数信息表 StationInfo,通过台站号(iiiii)和台站类型(Statype)联合两个字段作为唯一标识主键可区分所有在用的台站,完成对各类气象台站参数信息元数据进行核实校对数据库数据导入,由于台站参数种类较多,目

前只选取常用主要的台站参数,在日常工作中台站参数发生变更,大部分是因为台站迁移而导致的如经纬度等一些物理量纲参数的变更,所以对于台站参数元数据表字段类型需要分类定义,例如经纬度参数定义为 Numeric (8,3),所有高度参数定义为 Numeric(6,1),其他参数定为 Varchar,以上设置为以后的 SQL 更新操作时消除定义分歧,在确定了主要参数字段之后,如果后期有新增的台站参数,采用末位追加方式添加新增台站参数字段的方式。表 1 为台站元数据参数信息表 StationInfo 结构。

表 1 台站元数据参数信息表 StationInfo 结构

字段名称	字段名	类型
台站号	iiii	varchar(5)
台站类型	Statype	varchar(20)
省	provnice	varchar(20)
市	city	varchar(20)
县	county	varchar(20)
站名(简称)	station	varchar(50)
站名(全称)	station_name	varchar(50)
地址	address	varchar(50)
地理环境	geoenvironment	varchar(50)
纬度	latitude	Numeric (8,3)
经度	longitude	Numeric (8,3)
纬度(度分秒)	latitude_dfm	Numeric (8,3)
经度(度分秒)	longitude_dfm	Numeric (8,3)
观测场海拔高度	altitude	Numeric (6,1)
气压感应器海拔高度	altitude_p	Numeric (6,1)
往后顺序添加参数字段...

2.2 台站元数据表与目标台站参数信息表 Json 格式设计

综上分析,拟采用 Json 格式制作外部配置文件,作为软件同步系统进行数据库同步操作的辅助载体,分别对台站信息管理系统所辖的台站元数据参数信息表(以下称元数据表)和各个业务系统所辖的目标台站参数信息表(以下称目标表),将其各自的台站参数信息数据库表名称、表字段以及数据库连接信息 3 类重要数据参数封装为 Json 格式的配置文件,如图 1 所示,它们以(key,value)的键值对形式存在,本文以元数据表与土壤水分业务系统的目标表为实例进行说明。

元数据表 Json 格式配置文件采用“元数据表名”作为其键值变量标识,“元数据表名”为“StationInfo”,然后根据数据库连接信息、表字段使用嵌套分组的

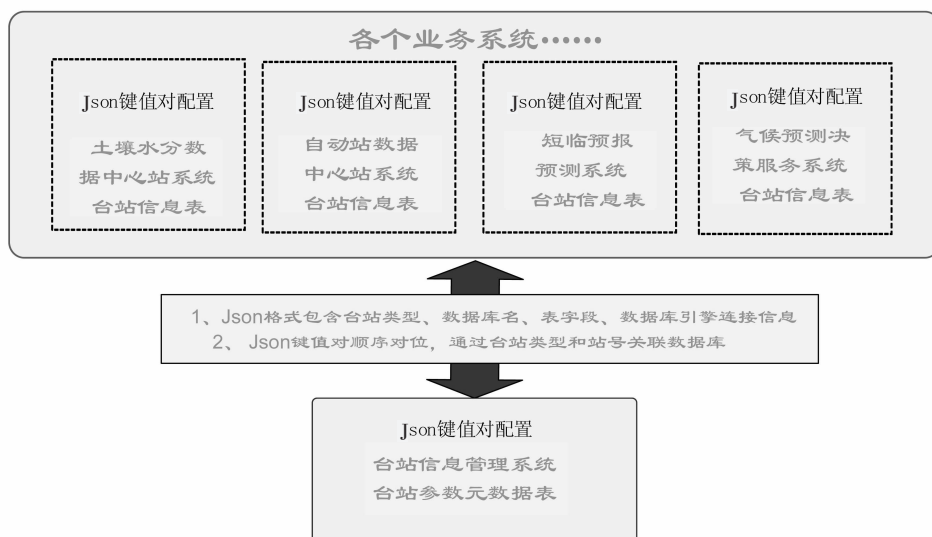


图1 元数据表与目标表 Json 格式关系示意图

方式编写 Json 配置文件。形成的格式内容包括元数据表名、数据库连接信息、表参数字段 3 种关键信息,其中“Database”键值对表示其数据连接引擎信息。以下内容为元数据表包含上述 3 种关键信息编写的 Json 格式的键值对配置信息。

```

{"StationInfo": {"Database": "Provider=SQLOLEDB;
Data Source =192.168.110.180;Persist Security Info =
True;Password = xxxz/5851142; User ID = s; Initial
Catalog= StationInfo", "1": "iiii", "2": "statype", "3":
"province", "4": "city", "5": "country", "6": "station", "
7": "stalevel", "8": "address", "9": "geoenvironment", "
10": "latitude", "11": "Longitude", "12": "latitude_dfm",
"13": "longitude_dfm", "14": "altitude", "15": "altitude_p",
"16": "height_windsensor", "17": "height_windplatform",
"18": "observmethod", "19": "stationclass", "20":
"builtime"}}
  
```

土壤水分目标表 Json 格式采用“目标表名_气象台站类型”作为其键值变量标识,“AGME”标识为土壤水分业务系统(还有如地面 SRUF、高空 UPAR 等其他业务系统),“目标表名_台站类型”为“Station_AGME”,设计 Json 格式时需要找出目标表中与元数据表定义相同的参数字段,并将其 Json 键值对设置的排序与元数据表对应字段的设置顺序保持一致,用“false”表示目标表字段顺序位上没有与元数据表定义相同的参数字段,通过这样的方式可以让元数据表与目标表参数字段键值对数目相等并且顺序对位,目的是为了当元数据表参数变更时快速完成对目标表字段的精确匹配定位。以下内容为土壤水

分目标表包含上述 3 种关键信息编写 Json 键值对配置信息,需要注意的是目标表字段使用中文命名。

```

{"Station_AGME": {"Database": "Provider=SQL-
OLEDB;Data Source =172.22.96.90;Persist Security
Info =True;Password = awssuser;User ID = awssuser;
Initial Catalog= hyaws", "1": "区站号", "2": "false",
"3": "省市名称", "4": "地区名称", "5": "false", "6":
"站点名称", "7": "false", "8": "站点地址", "9":
"false", "10": "纬度", "11": "经度", "12": "false",
"13": "false", "14": "观测场海拔高度", "15": "气压
传感器海拔高度", "16": "false", "17": "false", "18":
"false", "19": "是否有人站", "20": "false"}}
  
```

3 台站参数同步流程设计

当元数据表中某类气象数据的某个台站某个参数信息发生变动时,能够准确定位元数据表与目标表的参数字段,确认修改数值,先完成元数据表对应参数字段需要修改的参数值,然后生成与目标表对应参数字段匹配的 SQL 语句,最终完成对目标表对应参数信息的修改,流程简捷操作简单是整个参数同步流程的设计要求。以变更元数据表土壤水分台站纬度参数信息为例,来说明在此同步机制下如何完成对土壤水分系统中的目标表的数据同步操作流程,具体如下:

流程步骤 1:Json 配置文件导入。系统进入台站参数显示界面时,通过输入站号和台站类型,完成元数据表与目标表 Json 配置信息初始化,读取分解后

导入至缓存数组中。

流程步骤 2: 元数据表与目标表字段通过步骤 1 完成两边字段的顺序匹配映射并匹配生成 SQL 查询语句。参数的变更操作首先是进行参数信息查询操作, SQL 查询语句采用“目标表字段 A as 元数据表字段 A”的列别名方式拼接合成 SQL 查询语句, 通过对 Json 拆分后的参数字段按顺序拼接为列别名形式的 SQL 查询语句, 其中元数据表 SQL 查询语句如下, 其中“AGME”为土壤水分标识, 与站号一起通过前台查询参数传递进来。

```
select iiii as iiii , statyp as statyp , province as province, city as city, station as station, address as address, latitude as latitude, longitude as longitude, altitude as altitude, altitude_p as altitude_p, observmethod as observmethod from StationInfo where iiii = '59034' and statype="AGME"
```

对于目标表本身不需要标识台站类型, 元数据

表中的参数字段“statype”在目标表中为缺失字段, 为了保持元/目标表的字段顺序对位要求, 上述内容已采用“false”作为键值标识, 对于其他缺失字段以此类推, 得出目标表 SQL 查询语句如下:

```
select 区站号 as iiii , "false" as statype, 省市名称 as province, 地区名称 as city, "false" as country, 站点名称 as station, 站点地址 as address, 纬度 as latitude, 经度 as longitude, 观测场海拔高度 as altitude, 气压传感器海拔高度 as altitude_p, 是否有人站 as observmethod, from Station where 区站号='59034'
```

通过使用“as”列别名方式生成的 SQL 查询语句可以保持后台数据绑定代码和系统网站的显示标签的唯一性, 做到元/目标表参数字段增减变更与后台代码的低耦合无关联, 以下图 2 为数据库表查询流程示意图。

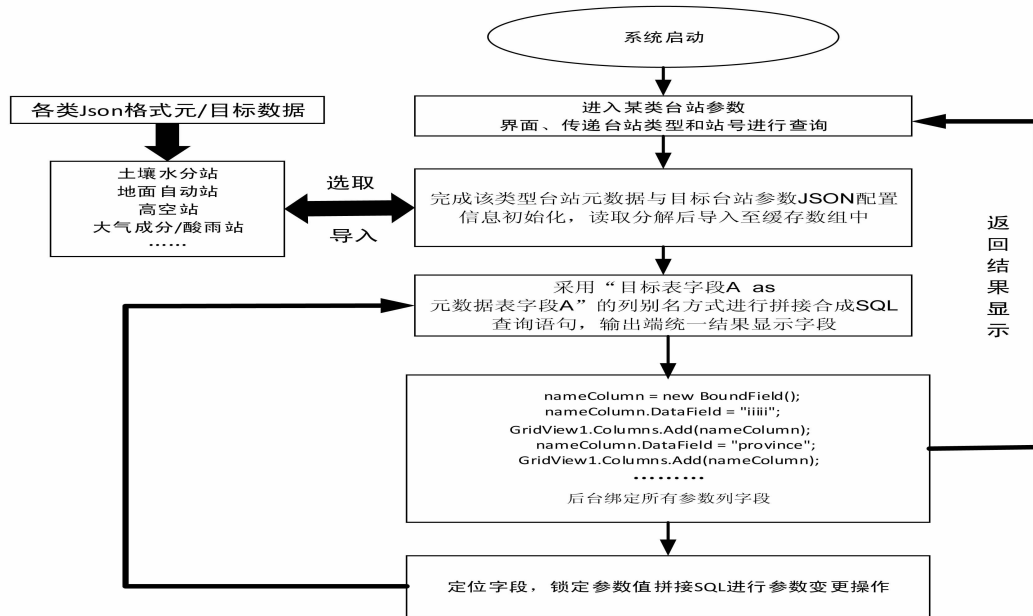


图 2 数据库表查询流程示意图

流程步骤三: 定位变更参数字段和变更参数数据, 自动生成 SQL 语句并完成数据更新操作。当元数据表中某个台站参数数据有更新变动时, 能够从元数据表字段快速定位需要更新的目标表字段, 传递修改参数值, 通过已知参数字段和变更参数数据

即可生成对应的 SQL 操作语句, 结合数据库引擎连接信息后完成元数据表与目标表对应字段的参数值的同步修改操作。以变更元数据参数字段“Latitude”参数值为例, 以下图 3 为“Latitude”参数数据同步变更操作示意图。

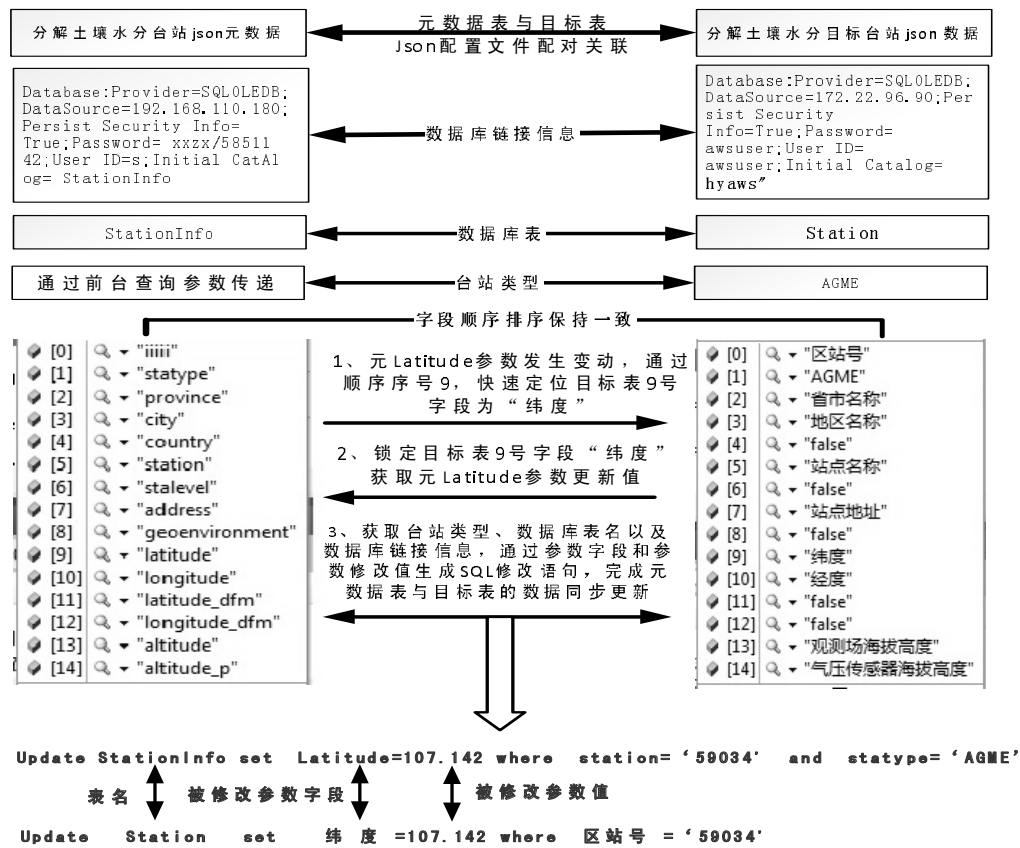


图3 “Latitude”参数数据同步变更操作示意图

4 实际应用效果

根据上述3个设计步骤,能够从根本上简化开发架构,淡化通常意义上所说的数据访问层(DAL层)、业务逻辑层(BLL层)、表现层(UI层)三层架构,减少代码规范性的要求从而降低开发难度,减少开发周期并快速完成系统建设;其次是实现系统代

码轻量化与鲁棒性,与后台代码松耦合状态使得内部代码变动较少并且保持较高的一致性,降低了应用系统后续升级维护、推广部署和迁移的难度。本文从触发方式、开发成本、开发难易程度等几个方面对几种同步方式进行了比较,具体各项指标如表1所示。

表1 3种数据同步方式各项信息指标

同步方式	触发方式	开发维护成本	开发难易程度	同步效率
Json方式	直接触发	很低	1000行以内	秒级别响应
手工SQL修改方式	间接触发	无	实际编写SQL语句	以分钟累计
商业定制系统	直接触发	高	10000行以上	秒级别响应

5 结语

采用自定义Json格式完成对元数据表与多目标表之间的数据同步流程,有如下几个优点。

(1)简化数据库管理和操作流程。采用Json配置文件的方式可以减少对元数据库端大量配置信息的设置,减少对数据库资源的占用;通过修改不同

业务系统对应的Json键值对配置文件即可完成对元数据表名、数据库连接信息、元数据表参数字段3种关键信息变更,减少对数据库操作和依赖。

(2)便于系统管理和维护。仅需要较少的Json配置文件即可完成对关键业务系统(约10几个)所辖目标表的关键信息的关联设置,由于是系统外配置文件,降低了管理人员对数据库操作和知识储备

的要求, 系统维护工作的重心主要是对元数据表和目标表 Json 键值对配置文件的制作和校对。

(3)可拓展性强。对于众多中小企业广泛存在的轻量化异构数据库表的数据同步需求, 通过自定义 Json 配置文件, 并且可以根据实际需求增加配置信息, 在较小的开发成本下即可完成系统数据同步功能的设计和修善, 并结合当下主流消息中间件技术, 提升系统同步技术支撑环境和效率以应对复杂的网络结构, 具有很好推广应用效果。

参考文献:

[1] 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): 257-26.

[2] 于卫国, 陈泽瀛, 文黎明. 基于 Json 数据交换模型的实时支付系统设计和实现[J]. 微型机与应用, 2016, 35(12): 84-87.

[3] 李伟程. 基于 Json 技术的特种设备信息公示系统的研究[J]. 信息通信, 2017(6): 275-277.

[4] 谷方舟, 沈波. Json 数据交换格式在异构系统集成中的应用研究[J]. 铁路计算机应用, 2012, 21(2): 1-4.

[5] 高静, 段会川. Json 数据传输效率研究[J]. 计算机工程与设计, 2011, 32(7): 2267-2269.

Research on synchronization strategy of small heterogeneous database based on Json

Huang Zhi, Li Tao, Song Yao, Su Chuancheng

(Guangxi Meteorological Information Center, Nanning Guangxi 530022)

Abstract: In order to solve the problem of how to quickly complete the data synchronization change among the parameter tables of different stations under the condition of different table fields in different meteorological business systems, this paper proposed a small heterogeneous database synchronization scheme. By encapsulating the database table name, table field and database connection information as Json format key value pair configuration file, SQL column alias was used to complete the query of the database table. When the parameter information of the source data table changed, the corresponding table fields and the target table could be accurately located, bounded to their modified parameter values, and automatically generate SQL statements to complete the operation of adding, deleting and modifying by decomposing and matching the order of the Json key value pair. The results showed that the way of data synchronization based on Json can simplify the development framework and reduce the development cost effectively, and quickly establish the process of data synchronization of various heterogeneous database tables. The way of configuring Json files outside the system can reduce the coupling degree of background code, which is conducive to the management, operation and maintenance of the later system, which has good application value.

Key words: station parameters; synchronization; Json; column alias