

张思,王楚钦,赵梅珠,等. 云南“天擎”应用融入示例研究[J]. 中低纬山地气象,2023,47(1):101-106.

云南“天擎”应用融入示例研究

张 思,王楚钦,赵梅珠,番聪聪

(云南省气象信息中心,云南 昆明 650034)

摘 要:“天擎”是数算一体的气象大数据云平台,具备海量数据存储、全业务贯通、数据应用高效的能力,支持各类气象应用的“云”化融入,全面支撑“云+端”的气象业务模式。该研究以云南省降水量告警系统为例,分析了应用系统融入“天擎”的全套流程。首先从系统技术架构的设计着手,对照融入要求,确定需要融入的切入点;然后从数据采集、数据存储、产品加工、业务监控、业务应用等 5 个方面对系统融入流程进行具体研究;最后,针对开发过程中需要注意的关键问题进行技术总结,可为后续融入工作提供参考。研究结果表明,基于“天擎”的系统融入开发流程较为简洁快速,普通开发人员较易上手,整个融入过程能从系统技术架构的多个层面减少不必要的开发步骤,缩短开发周期。

关键词:“天擎”融入;告警系统;气象

中图分类号:TP311 **文献标识码:**B

A Sample Study of Integrating the Application System into Yunnan "Tianqing"

ZHANG Si, WANG Chuqin, ZHAO Meizhu, FAN Congcong

(Yunnan Meteorological Information Center, Kunming 650034, China)

Abstract: "Tianqing" is a meteorological big data cloud platform integrating data and algorithms, it has the abilities of massive data storage, full-service connection and efficient data application. "Tianqing" supports "cloud" integration of various meteorological applications. It provides comprehensive support of "Cloud + terminal" meteorological service architecture. In this research, taking the integration of yunnan precipitation warning system into "Tianqing" as an example, we analyzed the whole integration process. First, according to the integration requirements, the breakthrough points were considered from the design of system technical architecture. Then the integration of this system is studied from five aspects including data collection, data storage, product processing, transaction monitoring and business application. Finally, it also technically summarized the critical issues which arose attention in the integration development process. It could provide a reference for the subsequent integration work. Results show that the integration development process is concise and fast relatively, and it is very easy for ordinary developers to get start. The whole integration process can reduce some unnecessary development steps and shorten the development cycle from multiple levels of the system technology architecture.

Key words: "Tianqing" integration; warning system; meteorology

0 引言

为适应气象业务发展需求^[1],气象部门构建了以气象大数据云平台(“天擎”)为“云”、气象业务

系统为“端”的“云+端”业务技术体系。“天擎”是数算一体的支撑平台,其数据全面,云上计算能力可直接支持应用。云南“天擎”于 2020 年 12 月投入业务试运行。按照中国气象局信息集约化的要

收稿日期:2022-01-06

第一作者简介:张思(1990—),女,工程师,主要从事气象信息技术工作,E-mail:cuit_zhangsi@163.com。

通讯作者简介:王楚钦(1989—),女,工程师,主要从事气象档案管理工作,E-mail:751306440@qq.com。

资助项目:云南省气象局自筹科研项目(2022 年):“天擎”地面国家站小时日值完整性和可用性分析。

求,气象行业内几乎所有的在用系统都必须逐步完成融入“天擎”的改造,新建的系统必须融入“天擎”。

鉴于云南“天擎”上线时间不久,系统庞大,涉及子系统繁多,底层系统采用了新兴 IT 技术(分布式大数据技术、Stream 流式传输等),业务方要将应用融入“天擎”是一项非常繁琐的学习和调试过程。“天擎”由中国气象局开发,在全国各省统一部署,省级系统支撑本省业务应用。在融入研究方面,国家级“天擎”有 CIPAS—海洋业务的融入案例研究,但目前在省级“天擎”的融入研究较少^[2]。为引导用户快速将应用融入到省级“天擎”,亟需 1 套简洁明了的实例分析流程。用户可将该流程套用到自身系统,整个流程畅通后,用户只需注重自身算法和系统的研发,无需过多关注系统的数据、存储、服务、监控等一系列繁琐项目,既能大大减轻融入工作量,也使得用户更愿意将系统融入“天擎”。

该研究以云南省降水量告警系统融入“天擎”为例,提出了 1 套适用于省级新增业务系统融入“天擎”的流程,整理了融入过程中的关键技术及一些需要特别注意的点,以期为省级业务用户后续融入研究工作提供参考。

1 系统介绍

“天擎”由交换及质控系统、产品加工系统、挖掘分析系统、存储与服务系统、业务监控系统和基础设施资源等 6 大系统组成,具备海量数据存储和在线加工能力,在数据、算力等方面可支撑应用云化融入,不断融入业务算法、业务产品和业务应用。

云南省现有 125 个国家气象站,3200 多个区域气象站,台站 1 min 上传 1 次数据,面对庞大的数据量,气象资料的质控任务也日渐加重。降水量是最重要的 1 个气象要素,当发生强对流天气过程时,自动站的监测数据对在应急部门的决策具有非常大的参考价值,因此对该项数据的准确性和及时性要求较高^[3]。若降水量等气象要素达到行业内预定的某个阈值时,可以及时提醒值班人员做出相应的预警服务,大大提高工作效率并减少主观疏漏^[4]。本示例的降水量告警系统就是在此环境下应运而生的。该系统使用 Python 语言进行开发,系统的计算及存储资源依托于“天擎”平台,代码部署在“天擎”的加工流水线服务器上长期运行;系统的监控信息存储于“天擎”平台,实时存入 Elasticsearch(简称 ES)索引库,ES 是一个分布式、高扩展、高实时的搜索与数据分析引擎,具备 TB 级数据的

存储服务能力,非常适用于存储和分析海量日志及告警信息;系统的服务依托于企业微信公众号集成告警。

本文的融入研究涉及到一个集中监控平台,即“天镜”。“天擎”和“天镜”是并列关系,“天擎”平台的资源、系统、数据、流程等都由“天镜”集中监视和控制。融入“天擎”的系统也必须融入“天镜”进行集中监控。

2 融入技术架构设计

《气象信息系统集约化管理办法》气发[2018]117 号文件提出业务系统融入“天擎”的 4 个要求:数据集约、流程集约、平台集约和监控系统集约。根据要求理清融入的主要思路:(1)产品存入“天擎”; (2)算法纳入加工流水线; (3)交互功能服务化; (4)系统监控进“天镜”。融入开发方应从数据采集、数据存储、产品加工、业务应用、业务监控等 5 个方面进行融入考虑,系统总体技术架构见图 1。

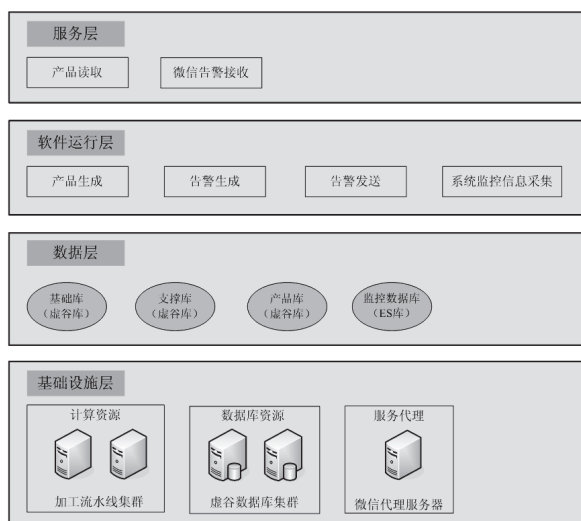


图 1 降水量告警系统总体架构

Fig. 1 Architecture of precipitation warning system

该降水量告警系统为多层软件体系架构(图 1),包含基础设施层、数据层、软件运行层和服务层 4 个层面。基础设施层包含计算资源、存储数据库资源(物理设备)和微信服务代理运行资源(虚拟机),是最底层设备,由“天擎”和“天镜”共同支撑,以实现平台集约。数据层包含该系统的输入、输出以及监控信息等相关数据,输入、输出产品数据存储于“天擎”的虚谷数据库,监控信息存储于“天擎”的 ES,以实现数据集约。软件运行层即该系统涉及的所有算法及单独运行的子程序,其中产品生成算

法和告警生成算法由“天擎”加工流水线提供计算资源,告警发送和监控信息采集由“天镜”提供代理,以实现流程集约和监控集约。服务层主要反映该系统实现的所有功能,产品读取由“天擎”虚谷数据库提供,微信告警服务由“天镜”告警模块提供,以实现业务应用融入,平台集约。

3 融入设计与实现

3.1 融入“天擎”的操作流程

本示例系统融入“天擎”的主要操作流程(图2)为:(1)申请账户及资源:在“天擎”portal 平台申请业务融入账户以及开发系统所需的资源(如存储、数据、算力等)。例如本示例申请的存储资源为5G 的虚谷数据库,数据资源为中国地面逐小时要素资料的读取权限,算力资源为2 核 CPU,4GB 内存和20GB 存储容量的 Red Hat Enterprise Linux 7.5 操作系统。(2)算法研发:以产品为粒度剥离算法,并基于“天擎”的存储技术进行算法研发。本示例研发了降水量累计和 EI 生成2 个算法。(3)算法部署:在“天擎”portal 平台部署并加载算法,配置算法运行策略(定时运行,数据源触发运行等)和产品的服务方式(使用 MUSIC 接口对外服务,或其他 Web 服务等)。(4)开发服务系统:从“天擎”获取产品进行系统所需的各项服务开发,本示例是基于自身产品(降水量累计产品)的微信告警提醒服务开发。(5)监控融入“天镜”:按“天镜”规定的标准将系统监控信息 DI 及告警信息 EI 存入 ES,并开发监控界面。

3.2 融入后系统工作流程

降水量告警系统的工作流程(图3)主要包含以下几个步骤:(1)首先以直连数据库的形式连接“天擎”系统的虚谷数据库,获取输入数据(逐小时降水量数据),对数据进行累计计算,得到的结果产品回写到虚谷数据库的产品表。(2)调用部署在“天擎”加工流水线的告警生成算法,生成告警信息 EI。

(3)连接“天镜”的 ES,将上一步生成的 EI 通过 REST 接口的形式写入,在“天镜”的告警平台(Alert)配置告警消息的生成。(4)部署微信代理平台,创建企业微信,配置告警过滤规则(MongoDB 数据库),将不同的告警信息推送至企业微信中对应的负责人。

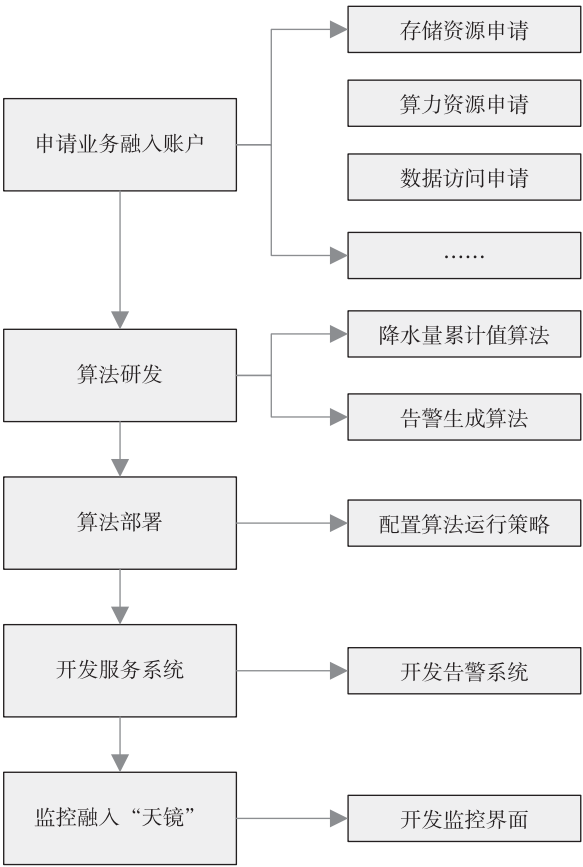


图2 业务系统融入“天擎”流程
Fig. 2 Flow chart of integrating service system into "Tianqing"

分析工作流程,第1 和第2 步实现了产品存入“天擎”以及算法融入加工流水线。同时,第2 步的 EI 生成算法实现了服务融入“天擎”。第3 步和第4 步实现了业务监控融入“天镜”。

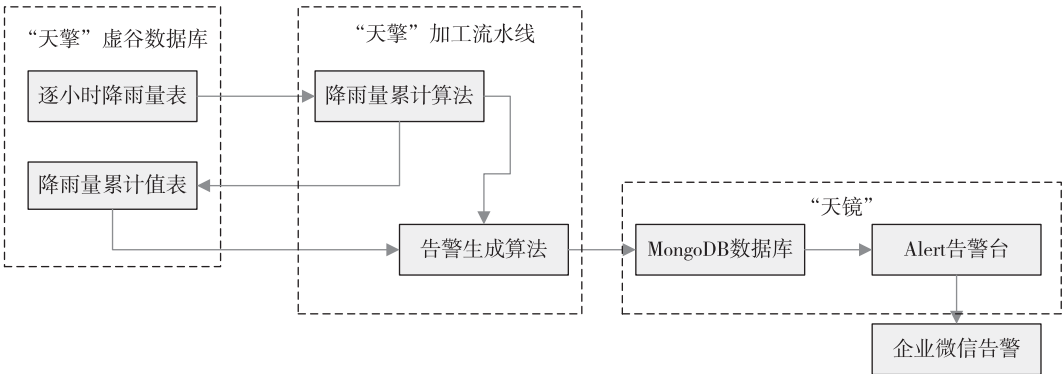


图3 融入后的系统工作流程
Fig. 3 Flow chart of Integrated system working process

3.3 融入关键问题

3.3.1 产品存入“天擎” “天擎”基于分布式技术建立了大数据存储体系,具备海量数据存储功能。系统开发前首先要针对产品选取合适的存储技术,“天擎”主要采用了 6 种存储技术:分布式表格系统、分布式关系型数据库、分布式分析型数据库、分布式文件库、分布式对象库和云数据库。本系统的降水量累计产品属于结构化站点要素数据,较适合分布式关系型数据库的存储模式^[5],其属于新增业务数据,需在“天擎”注册后规范化存储。因此,将此产品拟存储于个人申请的虚谷专题库中。按照存储模式设计降水量累计值产品的储存表结构(表 1),设计站号、资料累计时次以及累计小时数为主键。

表 1 降水量累计值存储表设计

Tab. 1 Design of cumulative precipitation storage table

字段名	字段类型	存储内容	是否主键
station_id	varchar	站号(字符型)	是
station_name	varchar	站名	否
lon	float	经度	否
lat	float	纬度	否
d_datetime	timeswap	资料累计时次	是
prf_n	integer	累计小时数(3、6、12、24 h)	是
prf	float	累计降水量(mm)	否

3.3.2 算法纳入加工流水线 “天擎”具备算法注册、发布、管理以及算法可视化编排等功能,结合任务调度策略,将算法转换为任务流水线运行。用户只需关注算法的研发、配置任务以及任务的管理,而算法运行需要的计算资源、数据资源、任务管理与监控、产品回写的存储资源由“天擎”提供。“天擎”提供多种任务调度方式,针对不同的任务调度,需要对算法进行相应的改造。

算法融入“天擎”的关键任务是对系统涉及的所有算法进行剥离,剥离标准一般为该系统产品的粒度。虽然本示例共生成 3 h、6 h、12 h 以及 24 h 降水 4 个产品,但这 4 个产品的加工由于输入数据源一致(都可由逐小时降水量数据加工得到)、存储方式相同(都可存储于分布式关系型数据库中),可以整合为 1 个产品即降水量累计值产品。另外,降水量超过设定的阈值时(阈值可自行设置,如行业规定的暴雨、大暴雨及特大暴雨量级阈值),会生成告警信息并发送至“天镜”ES 进行基于索引的存储,告警信息即为该系统的另一个产品。因此,本示例最终剥离出 2 个算法即降水量累计算法和告警生成算法。

算法剥离后即开始算法的研发,研发涉及 2 个

关键问题:算法输入输出及算法监控。输入输出原则为直接从“天擎”获取数据,中间数据和产品存入“天擎”的存储系统。“天擎”提供 2 种数据存取方式:直连数据库和接口调用。本示例 2 个算法都采用直连数据库的方式存取数据。降水量累计算法从虚谷库获取逐小时降水量数据,经累计计算后将结果写入用户申请的专题库,图 4 为降水量累计算法关键代码。降水量超阈值的告警信息由于产品的特殊性,类似于业务监控信息,较合适“天镜”ES 的存储方式,因此告警生成算法的输入数据为降水量累计产品,输出数据为 EI 告警信息,通过 REST 接口^[6]形式写入 ES,图 5 为告警写入的关键代码。此外算法参数的设置更多是采用外放的方式,而不是配置文件。

```
#计算过去n小时降水
def cacu_PRF_n(station,PRF_pre_n,n_time,n):
    PRF_pre = np.array(PRF_pre_n)
    st_all_prf_n = []
    for st in station:
        print(st)
        prf = 0.0
        for i in range(0, PRF_pre.shape[0]):
            if PRF_pre[i][0] == st:
                prf += float(PRF_pre[i][1])
        temp_prf = [prf,st,n_time,n]
        st_all_prf_n.append(temp_prf)
        #print(temp_prf)
    return st_all_prf_n
    # print(st)
    # break

station_ID = get_station()
for n in [1,3]:
    exec('prf_{} = {}'.format(n))
    n_time = get_time_pr(n)[0]
    p_time = get_time_pr(n)[1]
    prf_bu = get_PRF(n_time,p_time)
    exec('prf_{} = cacu_PRF_n(station_ID,prf_bu,n_time,n)'.format(n))
    sql = "insert into ZHANGSI.SURF_ESSEN_FAC_CACU_BEKHZS values(?,?,?,?)"
    cur.execute(sql,eval(f'prf_{n}'))
cur.close()
conn.close()
```

图 4 降水量累计算法关键代码

Fig. 4 Core code of precipitation accumulation algorithm

算法监控,即为按照监控需求将算法运行中产生的详细信息日志(DI)和错误告警信息(EI)按照规范^[6]进行封装并发送至“天镜”的采集接口。DI 信息一般分为 2 大类:产品数据类 DI 和算法运行类 DI。产品数据类 DI 主要为产品生成详细记录信息,及数据全流程流转详细记录信息。“天擎”将算法转换为任务流水线运行后的运行 DI,可由“天擎”提供,只需要在算法注册时勾选发送 DI 的选项,而算法本身运行的 DI 由算法内部封装后发往“天镜”。算法运行或产品生成遇到错误需要提醒时,则需要封装对应的 EI 发送至“天镜”。

```
#判断入库台站数,若<3300说明入库不正常.
if st_num_sod[0] < 3300 :
    kevent = f"[入库异常]当前时次入库数偏少:{st_num_sod[0]}!\n"
    print(kevent)
    p_EI2 = EventInformation()
    org_time = time.strftime("%Y%m%dT%H%M%S", time.localtime(time.time()))
    p_EI2.ORG_TIME = org_time
    p_EI2.fields.update(GROUP_ID='0P_BEKMS_C',KEvent=kevent, EVENT_TYPE='0P_BEKMS_C-1-10-01', EVENT_LEVEL=\
        '2', EVENT_TITLE = '资料入库环节异常',KObject= '资料入库环节异常' )
    pp_EI2 = p_EI2.__dict__
    EI2 = "[" + json.dumps(pp_EI2)+ "]"
    print(EI2)
    send_request_ei(json.loads(EI2))
#发送EI到天镜
#获取世界标准时间
elif prf_buf2.size != 0:
    utc_t = prf_buf2[0][1]
    if utc_t.hour+8 >= 24:
        hour = utc_t.hour + 8 - 24
        day = utc_t.day + 1
    else:
        hour = utc_t.hour + 8
        day = utc_t.day
    kevent = utc_t.strftime('%Y{y}%m{m}{d}{dv}{h}').format(y='年', m='月', d=day, dv='日', h = hour) + \
        '时台站降水量超50mmtop10:\n'
    for n in range(prf_buf2.shape[0]):
        if n == 10:
```

图 5 告警生成算法关键代码
Fig. 5 Core code of alarm generation algorithm

3.3.3 交互功能服务化 该示例生成的数据产品,由“天擎”portal 平台配置管理读写接口,通过 MUSIC 接口提供对外服务。“天擎”MUSIC 子系统已经与“天镜”进行了监控对接,服务监控信息可直接从“天镜”获取。MUSIC 沿用原来 CIMISS 时期的标准,读写使用方便,若现有的 MUSIC 接口无法满足业务需求,“天擎”还提供了接口众创服务,用户可按照接口规范开发自己的众创服务接口。

3.3.4 系统监控进“天镜” 业务监控的关键问题是 DI 和 EI 的封装和发送,需在算法设计阶段考虑进去。本示例监控分为 4 个层面:基础资源层、软件运行层、数据层和服务层。其中,基础资源层包含服务器性能、存储系统、网络流量等指标,这些指标“天镜”都能通过 agent 采集代理获取,用户需要为其配置相应的应用存储监视目录。软件运行层,“天镜”提供数据库 (oracle、mysql)、中间件 (Redis、tomcat 等)、应用进程、应用服务端口等的监视,本示例系统需要监控的是 2 个算法的运行情况和微信代理端口的服务情况。数据层的监控,“天镜”提供实时资料流程及 MUSIC 数据访问监视,并提供数据 DI、运行 DI 及告警 EI 的存取服务。服务层需监控业务服务指标、页面访问状态指标、产品通过 MUSIC 提供数据服务指标等。

监控页面的设计可基于“天镜”众创平台开发或自行开发,其关键问题是 2 种开发方式监控信息的来源都必须是“天镜”。该系统的监控界面基于“天镜”众创平台开发,服务层仅监视 1 项指标,即

降水量累计值产品通过 MUSIC^[7] 提供服务的指标;数据层监视降水量累计值产品的完整性和当前时次资料流程;软件运行层监视累计算法的运行情况和专题库的数据库状态;基础资源层监视算法运行所在的服务器性能、微信代理程序部署所在的机器性能以及网络连通状态等。

4 融入效果

4.1 产品存储融入效果

该示例系统仅生成 1 种数据产品(降水量累计值),存储于个人申请的专题数据库的虚谷数据库中,存储结构为分布式。数据库的存储性能、读写能力和容量都基于“天擎”平台自身的性能,由平台统一管理,不需用户维护。“天擎”面向应用的分布式存储模式使得其基础设施资源可横向扩展,保证了平台的高性能。业务开发用户可通过平台管理自己申请的数据,包括审核数据访问申请,配置数据服务接口等,其它用户可申请使用该产品。因此,存储融入后实现了产品共建共享,达到数据集约的目的。

4.2 算法融入效果

算法融入“天擎”的加工流水线后,直接由平台管理运行。算法触发提供 3 种方式:定时触发、顺序触发和数据源触发,管理较为方便。业务开发用户仅需关注算法本身及配置运行策略,无需过多关注算法运行所需的硬件资源。加工流水线实行算法确权共享,其他“天擎”用户若要使用该算法可提交

申请,经算法所有人审核后开通使用权限。可避免重复开发,提高开发效率,节约大量的人力物力。

4.3 服务融入效果

本系统的服务为降水量微信告警提醒服务^[8],用户关注企业微信订阅号^[9],即能接收告警,告警信息取自“天镜”,通过微信代理推送至企业微信,告警内容信息由“天擎”加工流水线上部署的告警生成算法自动生成。

4.4 业务监控融入效果

基于“天镜”众创平台的界面开发采用简单脚手架的方式进行搭建,功能实现可使用控件拖拽配置的形式,开发较容易上手。通过 REST 接口形式从“天镜”的 ES 中获取实时监控 DI,时效性较高,不易出错。相比于传统的关系型数据库,ES 对海量气象监视日志数据存储较为友好^[10],大日志数据实时搜索响应效率较高^[11]。该系统的监控界面包含基础资源层、数据层、业务运行层和服务层 4 个层面的监视。界面开发完成后可打包成产品模块部署于本省“天镜”平台,实现监控集约化。

该系统自上线以来,经过不断调试,告警信息的推送几乎未出现漏发和错发的情况,并且告警从生成到企业微信公众号收到能达到秒级的时效性。算法运行方面,算法部署于“天擎”的加工流水线,自“天擎”上线以来,从未出现过平台故障,运行效率也较高。

5 结论

该降水量告警系统生成的降水量累计产品是气象行业较为常见的数据,系统开发的技术架构设计思路和产品在“天擎”的存储方式等,都可以被其它产品生成类的系统开发工作借鉴。该系统融入“天擎”的流程,涉及数据存取、算法融入、服务融入以及业务监控融入 4 个关键融入要求,是 1 套较为完整的融入方案,对其它类似数据产品生成系统的融入具有极强的参考价值。

所有应用融入“天擎”的工作均可从以下 4 个方面来考虑:

(1)数据存取。主要考虑应用系统输入数据取自“天擎”,中间和结果产品存入“天擎”。数据融入后,可提高读写速率,方便统一管理。

(2)算法纳入加工流水线。主要关注算法的研发,运行任务的配置和管理以及算法运行中 DI、EI 的采集和封装。算法需按产品粒度进行剥离,不受开发语言及环境的限制。

(3)服务融入“天擎”。应用生成的产品可由“天擎”配置读写接口,通过 MUSIC 接口提供服务。在现有接口无法满足应用需求时,可按规范自行开发。

(4)系统监控纳入“天镜”。主要关注监控信息的采集和存储。应用系统的基础设施运行、算法运行、数据流程、应用服务等流程产生的 DI/EI 均应经采集封装后发送至“天镜”的 ES 存储。业务开发方只需按标准封装和发送 DI/EI,无需考虑数据的存储机制、存储基础设施资源以及数据库的维护。

综上所述,整套融入流程简洁明了,减少了融入过程中诸多管理方面的开发,能节约开发资源,缩短开发周期。

参考文献

- [1] 叶青,陆曼曼,林明.大数据时代气象信息化面临的问题与对策探讨[J].中国设备工程,2021(14):230-231.
- [2] 冯勇,李微,朱辉,辛文鹏.云计算环境下山东省气象大数据云平台的设计与实现[J].信息技术与信息化,2021(5):147-150.
- [3] 田红卫,任琦,钟延琴,等.榆林市气象防灾减灾服务系统的设计与实现[J].中低纬山地气象,2021,45(4):114-116.
- [4] 安承德,杜小玲,郎贵英,等.石阡 2020-05-20 强降水天气气象服务经验总结和思考[J].中低纬山地气象,2021,45(6):94-98.
- [5] 徐拥军,何文春,刘媛媛,等.气象大数据存储体系设计与实现[J].电子测量技术,2020,43(22):19-25.
- [6] 曾乐,孙超,张来恩,等.基于大数据技术的气象业务监视数据采集处理[J].计算机仿真,2021,38(7):181-188.
- [7] 赵冰燕,郭彩莲,来志云.基于青海气象大数据云平台的数据服务接口[J].青海科技,2021,28(1):82-86.
- [8] 李四维,韩波,张敏,等.地市级气象综合观测自动化监控报警系统的设计与应用[J].中低纬山地气象,2021,45(6):116-120.
- [9] 李正,陶鑫,黎斌,等.基于企业微信的网络故障告警系统应用与研究[J].科技视界,2019(19):56-57.
- [10] 孙超,肖文名,曾乐,等.海量监视数据云存储服务模型的设计与实现[J].武汉大学学报(信息科学),2020,45(7):1099-1106.
- [11] BAI J. Feasibility analysis of big log data real time search based on Hbase and ElasticSearch[C]. International Conference on Natural Computation(ICNC),2013(9).