

一种云原生数据智能服务生产平台设计与实现

王浩田¹, 鄂海红¹, 王 勇², 宋美娜¹

(1. 北京邮电大学 计算机学院(国家示范性软件学院), 北京 100876;

2. 联洋国融(北京)科技有限公司, 北京 100088)

摘 要:将数据智能服务嵌入到实际业务场景中创造价值,已成为众多企业数字化转型的重要特征。针对数据智能服务落地因过程多且复杂而导致的开发周期长、成本高等问题,设计并实现了一种数据智能服务生产平台。该平台底层基于 Kubernetes 容器云并以容器的方式对外提供了计算、存储、虚拟化和监控等基础服务,中间层基于这些基础服务搭建了用于海量多源异构数据统一组织管理的数据仓库、用于自动化生产数据智能服务的生产线和用于细粒度复用已有服务生产新服务的引擎等基础设施,顶层按照不同的业务场景对这些基础设施提供的能力进行了封装并设计了数据管理与标注、算法模型研发、模型批量生产和数据智能服务管理等模块。实际运行结果表明,该平台有效提高了数据资产、模型资产、服务资产之间相互转变的自动化能力,缩短了服务开发周期、提高了服务开发效率并降低了人力成本的投入。

关键词:数据智能;云原生;大数据;人工智能;持续生产

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2023)10-0015-07

doi:10.3969/j.issn.1673-629X.2023.10.003

Design and Implementation of a Cloud Native Data Intelligent Service Production Platform

WANG Hao-tian¹, E Hai-hong¹, WANG Yong², SONG Mei-na¹

(1. School of Computer Science (National Pilot Software Engineering School), Beijing University of

Posts and Telecommunications, Beijing 100876, China;

2. Lianyang Guorong (Beijing) Technology Co., Ltd., Beijing 100088, China)

Abstract: Embedding data intelligence service into actual business scenarios to create value has become an important feature of many enterprises' digital transformation. Aiming at the problems of long development cycle and high cost caused by the multiple and complex process of data intelligence service landing, we design and implement a data intelligence service production platform. The bottom layer of the platform is based on the Kubernetes container cloud and provides external basic services such as computing, storage, virtualization and monitoring in the form of containers. Based on these basic services, the middle layer has built a data warehouse for the unified organization and management of massive multi-source heterogeneous data, a production line for automated production of data intelligent services, and a service engine for fine-grained reuse of existing services to produce new services. The top layer encapsulates the capabilities provided by these infrastructures according to different business scenarios and designs modules such as data management and annotation, algorithm model development, model mass production and data intelligent service management. The actual operation results show that the platform effectively improves the automation ability of the mutual transformation among data assets, model assets and service assets, shortens the service development cycle, improves the service development efficiency and reduces the input of labor cost.

Key words: data intelligence; cloud native; big data; artificial intelligence; continuous production

0 引言

随着数字技术革新与发展,数字化转型已成为企业转型升级的必由之路^[1]。数据智能业务通过“记录—统计—分析—预测—决策”的方式为企业在不同

场景的运营与发展提供了核心资料支撑,是企业实现数字化转型的重要目标之一。传统企业的数据智能业务发展对于当前过载的互联网、大数据环境显得力不从心,数据说话、精准定位、快捷响应已经成为现阶段

收稿日期:2022-12-04

修回日期:2023-04-05

基金项目:国家自然科学基金(62176026)

作者简介:王浩田(1997-),男,硕士研究生,通信作者,研究方向为大数据及人工智能技术、云原生技术;鄂海红(1983-),女,博士,教授,研究方向为大数据及人工智能技术、分布式微服务架构。

的基本诉求。

虽然国内外研究人员在大数据计算平台、AI 训练平台以及服务开发平台^[2-7]等方面都进行了大量的研究和实践,但是都集中在以 DevOps 或 MLOps 的软件生命周期方法单一地对大数据处理流程和 AI 训练流程进行优化,仍无法满足数据智能业务快捷开发的需求,存在以下问题和挑战:

(1)海量多源异构数据集中管理与利用的问题。数据智能业务所需要的数据来源广且量大,不同的数据源所属领域不同,它们提供的数据呈分散、异构的特点,缺乏了统一的数据资源组织管理^[8-11]。

(2)服务开发过程依旧繁琐、开发周期依旧长的问题。数据智能业务的落地需要经过大数据存储与处理、模型设计与训练、模型性能评估、应用部署运维等过程,需要投入大量的人力分工协作,且不同工作间具有依赖性,对后期发现问题的返工修正的不断更新,导致更高的时间和经济成本^[12-13]。另一方面,前期研发所形成的算法资产,对于新的数据和特定的场景,都有持续训练、集成和部署的需求^[14]。

(3)服务之间的可重复利用问题。在数据智能场景的服务开发中,复用多个已构建数据智能服务的原子能力形成新的服务,是一种提高生产力的有效方式^[15]。但服务之间的依赖关系、重复调用导致了更长的响应时间和系统资源占用量,如何设计出高效的组合服务显得尤为重要。

因此,该文旨在设计并开发一个云原生数据智能服务开发平台,实现从多源异构数据的统一组织管理,到数据智能服务自动化生产线的构建,再到高性能数据智能服务引擎的搭建,提高从数据资产到模型资产的价值转变能力,模型资产到服务资产的持续训练、集成和部署的自动化能力,以及服务资产之间的可重复利用能力,使相关人员能够通过该平台快速完成数据智能服务的开发与管理。

1 研究内容

针对以上问题与挑战,该文重点围绕数据智能服务自动化生产线和高性能分布式数据智能服务引擎两个方面进行研究与分析,并完成了一个云原生数据智能服务生产平台(见图1)。主要研究内容有以下三项:

(1)设计并实现了数据智能服务自动化生产线。该生产线实现了从多源异构的海量数据资产存储与管理,到大数据驱动的多版本 AI 模型资产的生产与管理,再到高性能数据智能服务资产的部署与管理的自动化生产流程,有效缩减了数据智能服务的开发周期,减少了人力投入,实现了降本增效的目的。

(2)设计并实现了高性能分布式数据智能服务引擎。一方面,引擎实现了数据智能服务实例根据负载状况自动扩缩容,提高服务的稳定性和响应速度;另一方面,引擎以表达式即服务的方式,实现了对数据智能服务原子能力的复用,并对表达式的构建与调用进行了优化,解决了服务编排时服务依赖问题带来的响应时间过长和服务重复调用导致的系统资源浪费的问题。

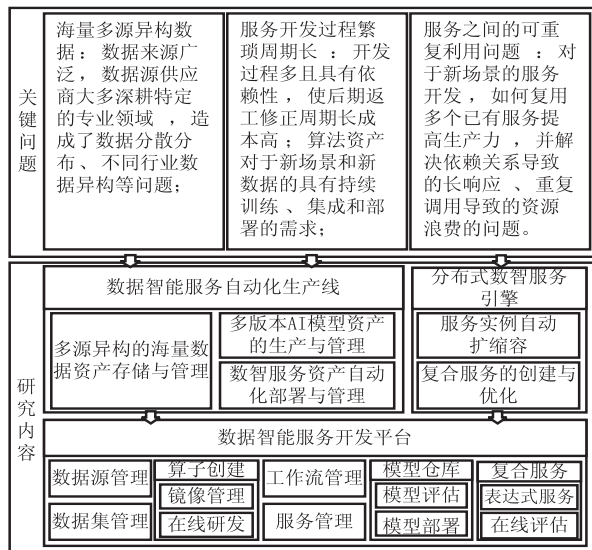


图1 关键问题与研究内容

(3)设计并实现了云原生数据智能服务生产平台。在产品方面,该平台将算法与业务分离,使算法工程师更多地聚焦于算法的深度和广度并承接更多场景的算法设计,使模型运营专注于数据集构建和模型训练并快速达到业务指标。在系统功能方面,该平台提供了多源异构数据统一管理、融合多种异构数据的数据集构建与管理、算子在线开发与版本管理、图像化工作流拖拽开发与版本管理、模型评估部署与版本管理、服务健康监测、表达式服务在线编辑与测试功能。通过产品层面的解耦合与功能层面的自动化生产能力,大大提高了相关人员开发数据智能服务的效率。

2 数据智能服务自动化生产线

数据智能服务自动化生产线实现了从数据资产到模型资产再到服务资产的全方位管理与生产,以过程自动化的方式提高数据智能服务的生产效率,降低人力成本的投入。

图2是数据智能服务生产流水线的架构,流水线的运转依赖下层的基础服务层和基础设施层,并支撑了上层的系统应用层和场景应用。基础设施层在物理机集群上搭建了 Kubernetes 容器云,提供了容器运行的基本环境。以容器应用作为承载平台计算任务的基本单元,可以实现运维的自动化和弹性伸缩等功能并

最大限度地发挥物理机群的性能和资源规模调度的效率优势。基础服务层以容器应用的方式封装了 Spark、MapReduce 等计算服务, HDFS、Hive、Mysql、Minio 等存储服务, Docker Image、Docker Container 等虚拟化服务, Grafana、Prometheus 等监控服务, 为流水线的核心功能层提供了基本服务。核心功能层实现了数据管理、自动化模型生产与管理、服务引擎等流水线的核心功能。系统应用层对流水线的功能进行了封装, 实现了数据管理与标注、算法模型研发、模型批量生产和数智服务管理等应用模块, 为多场景数据智能服务的生产做支撑。

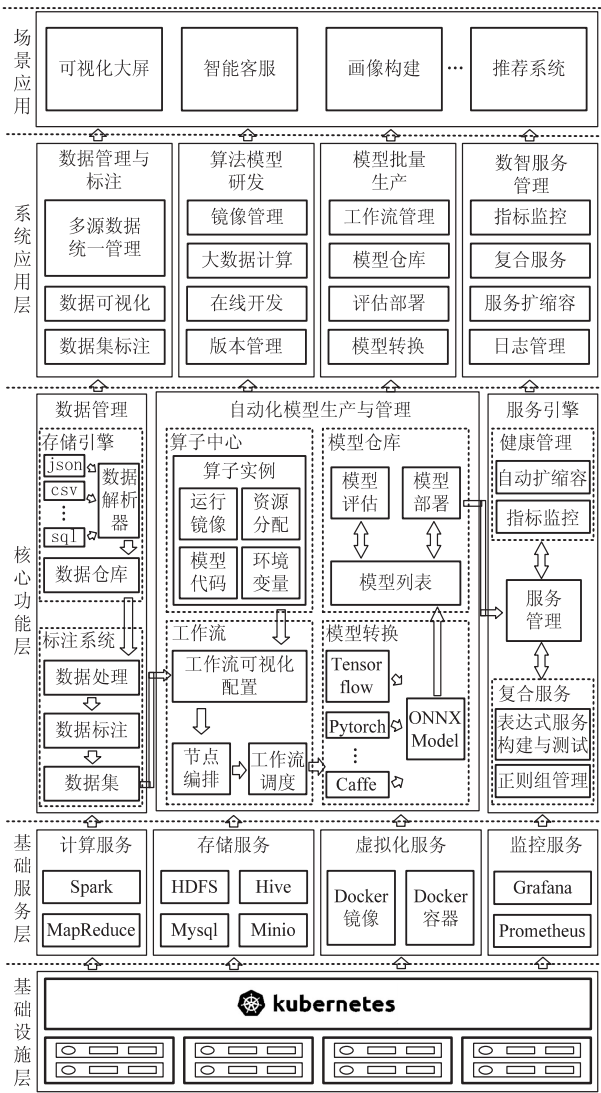


图2 数据智能服务生产流水线的架构

数据智能服务生产流水线为大屏可视化、智能客服、画像构建、推荐系统等多个场景应用的构建提供了智能服务支持。

2.1 数据管理

数据智能服务的落地需要海量不同来源的数据作支撑, 这些数据呈分散、异构的特点, 因此, 亟需一个统一的数据资源仓库实现对数据的统一管理。

图3是该文设计的数据资源仓库, 实现了对海量多源异构数据的存储与管理。当用户提交数据文件或数据源信息后, 数据仓库通过四个过程将其进行存储: (1) 将数据文件或数据源信息上传到 HDFS 中并根据配置信息创建一个文件导入任务; (2) 任务分类会将任务放入不同的管道中排队; (3) 空闲的解析器会在创建区根据它的类型装载不同的文件导入任务; (4) 装载完毕后, 在运行区解析文件并将数据插入到 Hive 仓库中。其中任务解析器是一个 Job 类型的 Pod, 当解析器运行失败 Kubernetes 会自动重启该任务, 直到解析器运行成功。当读取数据的时候, 可以执行相应的 Hive SQL, 并且可以实现多源异构数据的关联读取。通过编写不同的文件解析器, 将文件从 HDFS 转存到 Hive 中, 可以实现多源异构数据的统一管理。

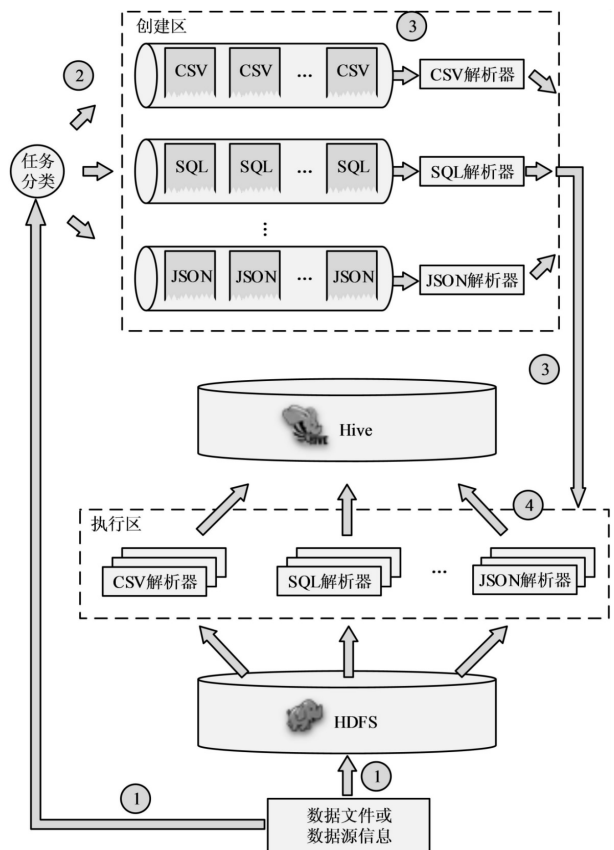


图3 数据资源仓库

图3中的 HDFS 和 Hive 是基于 Kubernetes 平台实现的大数据集群提供的存储服务, 采用这种方式部署大数据集群解决了传统的大数据集群参数配置复杂、运维难度大、缺乏自动修复能力、弹性扩缩容缓慢且无法及时随业务量变化等问题。

2.2 自动化模型生产与管理

数据智能服务开发过程多且复杂, 过程与过程之间具有依赖性, 后期的返工修复会导致更多的时间和经济成本, 以及在特定的场景和新数据的情况下, 都有服务持续生产的需求。因此, 该文在流水线中设计了

自动化模型生产 workflow 和模型仓库,实现了模型的自动生产与管理。

(1) 工作流基本运算单元——算子。工作流中的节点也称为算子,它是工作流运行的基本单元,对应 Kubernetes 中的 Pod。算子由代码文件夹(包含一个启动脚本和相关的程序代码)、运行镜像、运行参数等元素组成。

(2) 工作流调度引擎。将多个算子按照一定的依赖关系进行拼装组合,就可以形成一个自动化工作流。工作流调度引擎将会按照依赖关系,依次运行每一个工作流节点。已创建工作流资源可以输入不同的数据和设置不同的配置参数多次运行,生产出不同的模型或服务,实现算法资产的可持续集成、训练和部署。

图 4 是工作流调度引擎的执行过程,Web 端构造的工作流程图会放入管道中排队,空闲的工作流处理器依次读取管道中的工作流图进行解析与调度。工作流处理器维护一个集合,对工作流图进行层序遍历。每当运行完一个工作流节点,如果该工作流节点的后置节点不在集合中就将其加入集合,并对集合进行检查,将所有前置依赖运行完的工作流节点取出并将其作为 Job Pod 放入 Kubernetes 中运行,重复进行此操作,直到集合为空,工作流运行成功。对于工作流中的一个节点,会存在两次调度:一是等待全部依赖节点运行完成,才能够调度该节点运行;二是将节点调度到 Kubernetes 集群的节点运行,实现资源的负载均衡。

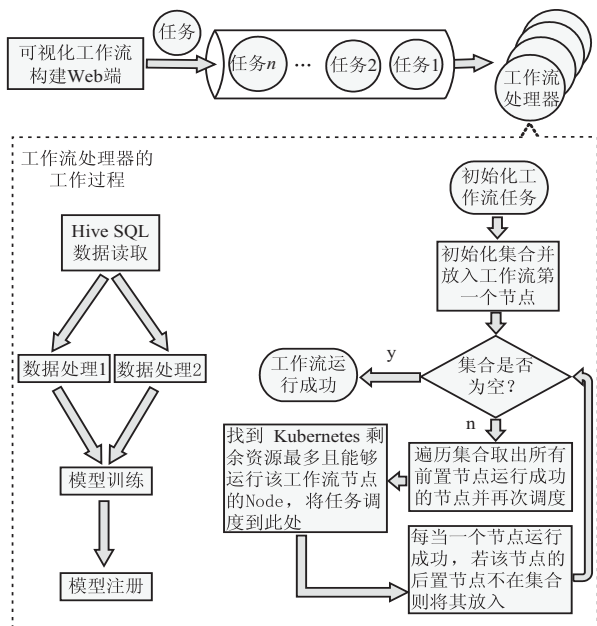


图 4 工作流调度引擎的工作过程

(3) 模型仓库。工作流中的模型注册算子会将训练完的模型存入到模型仓库,模型仓库会将实现模型资产的统一管理。

图 5 是模型仓库的架构,注册算子注入的模型首

先经过模型转换模块转换为 ONNX 模型,然后存入到模型库。模型库本质是一张 Sql 表,表中记录了模型的基本信息和模型所在 HDFS 中的路径。此外,模型仓库还提供了模型评估和模型部署的功能,模型评估通过在评估脚本模板中加载 ONNX 模型和测试数据集并与作为运行环境的镜像一起以 Pod 的形式在 Kubernetes 中运行,实现模型性能指标评估的计算。模型部署通过在推理脚本模板中加载 ONNX 模型并与作为运行环境的镜像一起部署为多个 Kubernetes 的 Pod 实例实现服务的多实例部署与注册。

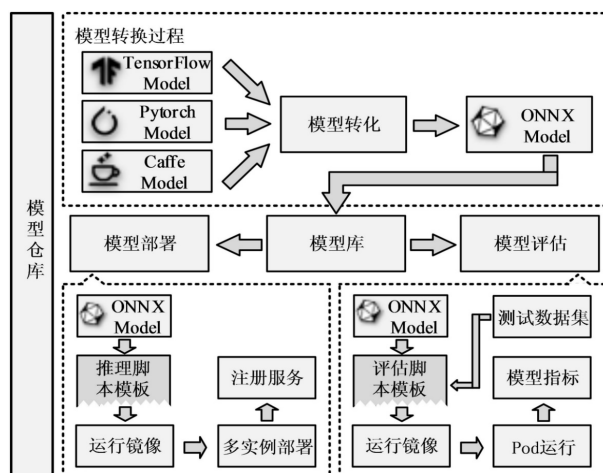


图 5 模型仓库功能与架构

2.3 服务引擎

模型仓库中创建成功的模型部署会将部署信息作为服务信息注册到服务引擎,服务引擎实现了对服务资产的统一管理。

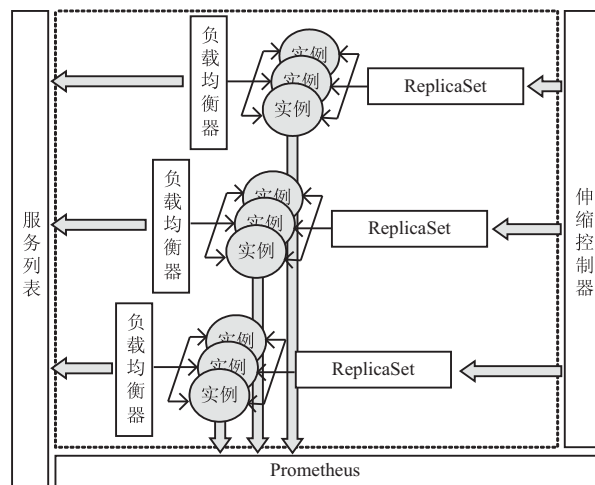


图 6 服务引擎架构

图 6 是服务引擎的架构,单个服务是由一个负载均衡器、一个 ReplicaSet 控制器和多个实例组成。服务列表绑定了每个负载均衡器的地址,通过服务列表可以查找到已经注册的全部服务。底层的 Prometheus 监控着实例的 CPU、内存等资源的利用率。伸缩控制器通过读取实例的指标进行判断,当某一指标超

过设定的最大阈值后,开始控制 ReplicaSet 控制器进行实例横向扩容;当所有指标均低于设定的最低阈值后,开始控制 ReplicaSet 进行实例缩容。通过这样的结构和工作过程,提高了服务在峰值时的稳定性以及减少了在低谷时的资源利用。

3 高性能分布式数据智能服务引擎

高性能分布式数据智能服务引擎以表达式即服务方式实现了对数据智能服务原子能力的细粒度复用,并对表达式服务的构建、调用等方面进行了优化,有效提高了服务开发效率以及服务响应速度和系统资源利用率。

数据智能服务引擎是三层结构(见图7):原子服务层、引擎中控层和应用层,底下两层采用了分布式的结构。核心层是引擎中控层,它采用 Leader-Worker 的结构保证中控层的高可用性和一致性,单节点的内部由功能决策、缓存、归并优化器、状态控制器、逻辑运算器、排序模块、存储服务、溢写模块等模块组成,实现了表达式的优化和表达式服务调用的优化。底层是原子服务层,它实现了 AI 模型的推理能力,由负载均衡器和多个服务实例组成。最上层是应用层,提供了表达式构建、表达式准确率在线测试和服务管理的功能。

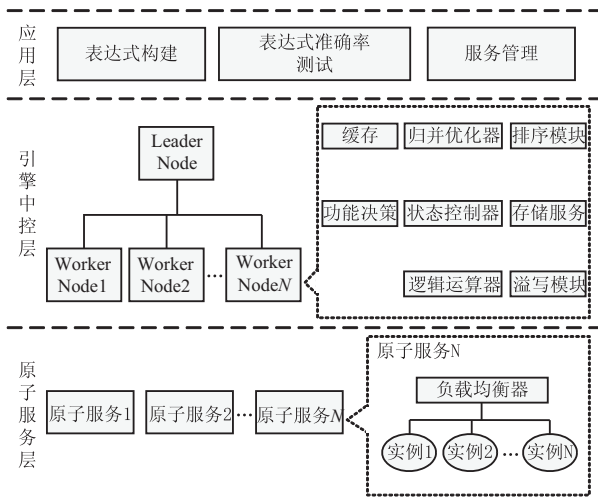


图7 数据智能服务引擎的三层结构

表1 引擎结构的主要模块

序号	模块名称	功能介绍
1	功能决策	根据请求类型,转发请求到不同模块
2	缓存	存储排序模块得到的热门预测请求,如果预测请求直接命中缓存,直接返回预测结果
3	归并优化器	对传入表达式提取公共原子服务进行归并,减少原子服务的重复调用次数
4	状态控制器	实现服务的可靠调用,当服务调用超时,重新发起调用请求,直到全部的原子服务调用完毕

续表1

序号	模块名称	功能介绍
5	逻辑运算器	根据表达式项(原子服务调用结果)的值,计算表达式的值,将计算结果存储到溢写模块,并返回结果
6	排序模块	定时根据服务ID和请求聚合历史运算结果并降序排列,将计算结果的前N项写入到缓存中
7	存储服务	提供分布式或其他类型的数据存储能力
8	溢写模块	当历史记录存储的总大小到达某一值,将数据写入到存储服务,避免经常写入带来的存储碎片和性能下降等问题

3.1 表达式服务

数据智能服务本质上是一种数据驱动的人工智能服务,不同类型的服务由多种原子能力组成,如文本多标签分类服务是由多种单标签分类原子能力组成,命名实体识别服务是由多种单实体识别原子能力组成。针对数据智能服务的这种特点,该文将多种不同服务的原子能力组合成逻辑表达式,以表达式即服务的方式实现了新服务的快速开发。

3.2 表达式服务的创建

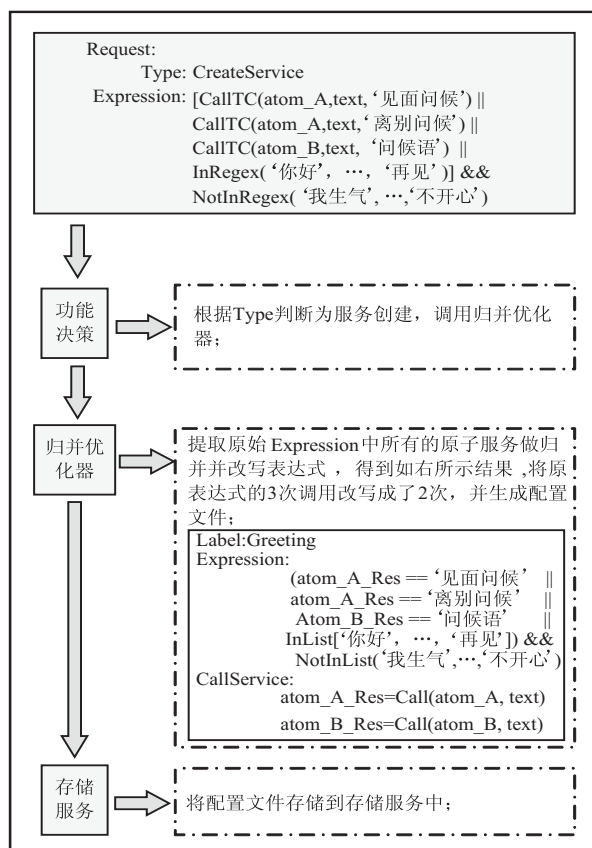


图8 表达式服务创建过程

图8展示了表达式服务创建的过程。功能决策模块首先会根据请求的Type字段判断请求为服务创建类型,然后请求转发给归并优化器。归并优化器会提取共有的原子服务进行聚合,示例中的表达式有三个

数据智能服务项(也叫做原子服务项),共两类 atom_A 和 atom_B。具体归并聚合例子为:[CallTC(atom_A, text, ‘见面问候’) || CallTC(atom_A, text, ‘离别问候’) || CallTC(atom_B, text, ‘问候语’) || InRegex(‘你好’, ..., ‘再见’)] && NotInRegex(‘我生气’, ..., ‘不开心’),通过提取项中的原子服务,得到 atom_A, atom_A 和 atom_B, 然后进行归并得到 atom_A 和 Atom_B, 对原表达式进行改写,得到:(atom_A_Res == ‘见面问候’ || atom_A_Res == ‘离别问候’ || Atom_B_Res == ‘问候语’ || InList[‘你好’, ..., ‘再见’]) && NotInList(‘我生气’, ..., ‘不开心’),该样例表达式在 AI 服务引擎中会将原来的 3 次服务调用优化为 2 次。最后归并优化器会将得到的优化结果以配置文件的形式存储到存储服务中,完成表达式服务的创建过程。

3.3 表达式服务的调用

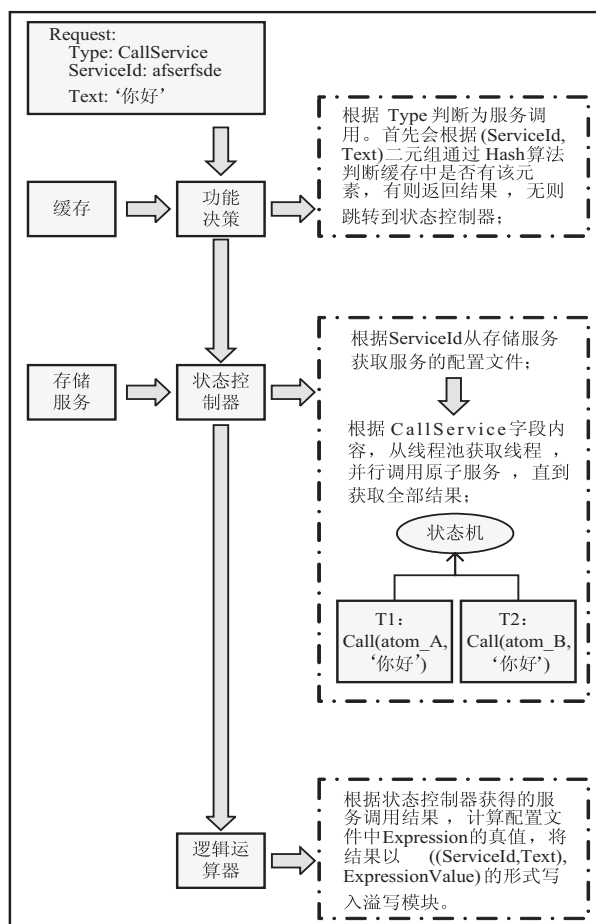


图9 表达式服务的调用过程

图9展示了表达式服务的调用过程。功能决策模块首先会根据请求的 Type 字段判断请求为服务调用类型,然后根据 (ServiceId, Text) 查找缓存中是否命中,如果命中直接返回结果,否则转发请求到状态控制器。状态控制器根据 ServiceId 从存储服务中找到表达式服务的配置文件,然后根据 CallService 中原子服务的

数量,创建等量的线程,并行调用原子服务,直到全部的原子服务返回结果,将结果传入到逻辑运算器。逻辑运算器根据原子服务的调用结果和正则列表,计算真值,将计算结果写入到溢写模块并返回给客户端。

3.4 缓存的更新

引擎中的溢写模块记录了所有请求与表达式服务的运算结果,当记录存储大小接近存储系统的文件块时,溢写模块会将这些记录写入到存储服务中。引擎中的排序模块会定期拉取历史请求记录以 (ServiceId, Text) 为 Key 进行聚合并降序排列,将结果的前 N 个写入到缓存中。

4 云原生数据智能服务生产平台

在数据智能服务自动化生产线和高性能数据智能服务引擎的基础上,该文设计并实现了云原生数据智能服务开发平台。一方面,该平台实现了数据资产到模型资产持续训练和集成的价值转变能力,模型资产到服务资产持续部署的自动化能力,以及服务资产之间的可重复利用能力。另一方面,该平台将算法与业务分离,使算法工程师更多地聚焦于算法的深度和广度并承接更多场景的算法设计,使模型运营专注于数据集构建和模型训练并快速达到业务指标。

4.1 平台的功能架构

图10是数据智能服务开发平台功能结构,该平台提供了数据管理、算子构建、模型开发、服务管理和复合服务5个模块,实现了对15种资产或资源的生产与管理。

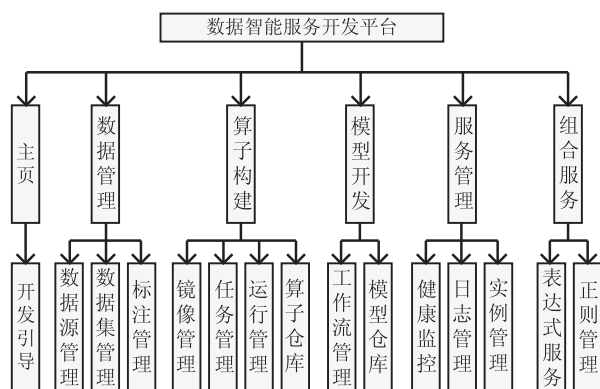


图10 平台功能结构

数据管理模块实现了对多种数据源和数据集的统一管理,其中数据集通过导入数据源并进行标注生产而来。算子构建模块实现了 Job、Task、Run、镜像、算子等资源的生产与管理,其中算子是一个 Run 的实例,Run 是 Task 通过输入运行参数运行的实例,Task 由镜像和项目脚本组成,Job 则是一系列 Task 的集合。模型开发模块实现了工作流、工作流运行历史、模型、模型评估、模型部署等资源的生产与管理,其中工作流是

对一系列算子编排而来, workflow 运行历史是 workflow 运行的实例, 模型则是通过 workflow 运行生产而来, 模型评估是通过输入测试集对模型性能进行评估的实例, 模型部署是将模型发布成服务的实例。服务管理实现了对服务实例的管理和监控, 其中服务实例是模型部署后产生的。复合服务实现了对正则组、表达式服务等资源的管理, 表达式服务由一系列正则组合和一系列服务实例通过逻辑运算组合而来。

表 2 数据智能服务开发平台的生产或管理的资产与资源

序号	模块名称	功能介绍
1	数据源	数据供应商提供的数据资产, 包括 csv、txt、sql 等多种不同的类型
2	数据集	由一个数据源或多个数据源按某些字段连接得到的中间表的某些列的数据制作而成
3	Job	Job 是对某些特定场景的 Task 的统一管理, 是一个 Task 集合
4	Task	由镜像和代码文件夹组成, Task 的配置包含了启动脚本和环境变量
5	Run	是 Task 添加运行参数并启动后的实例, Run 还实现了在线 Debug 功能
6	镜像	代码运行的基本环境, 实现了与外部的隔离
7	算子	某些运行成功的 Run 创建而来, 实质是代码文件夹、镜像、运行参数组成
8	workflow	对多个算子进行编排, 使其按一定的顺序运行
9	workflow 运行历史	workflow 运行的实例集合, 可以查看节点的运行日志
10	模型	workflow 运行过程中生产的产物
11	模型评估	用测试数据集测试模型性能产生的实例;
12	模型部署	将模型发布为服务产生的实例;
13	服务	平台的最终产物, 以 Http 接口的方式提供服务能力
14	正则组	一系列正则组合而来
15	表达式服务	由一系列正则和一系列服务通过逻辑运算组合而来

4.2 平台的技术架构

图 11 是平台的技术架构, 共分为 5 层: 基础设计层实现对 CPU 服务器、GPU 服务器等物理设备的统一管理; 虚拟化层利用 Docker Engine 和 Kubernetes 实现容器云, 用于分配、调度和管理 CPU 计算资源、异构 GPU 计算资源、存储资源、网络资源, 为上层设施提供基本计算能力和基本存储能力的支持。基础服务层提供 AI 训练服务、大数据服务和通用数据存储服务: AI 训练服务利用 Pytorch、TensorFlow、ONNX 等框架实现了 AI 模型的搭建、训练和部署等能力; 大数据服务利用 HDFS、Hive 实现大数据存储能力, 利用 Spark 实现

大数据计算能力; 通用存储服务利用 Mysql 实现关系数据存储, 利用 Minio 实现对象存储, 搭建镜像存储服务实现对镜像的存储。业务逻辑层将下层服务封装成功能接口供前端调用, 后端系统整体基础架构采用 Spring Boot 开发, 持久层框架采用 MyBatis, 实现对后端数据库进行 CRUD 操作, 前端与后端采用 Restful API 的方式进行通信。页面展示层使用 Vue.js 前端开发框架进行开发, 配合 Element-UI 样式组件库实现前端页面展示效果。同时通过 axios 异步调用的方式访问后端接口。

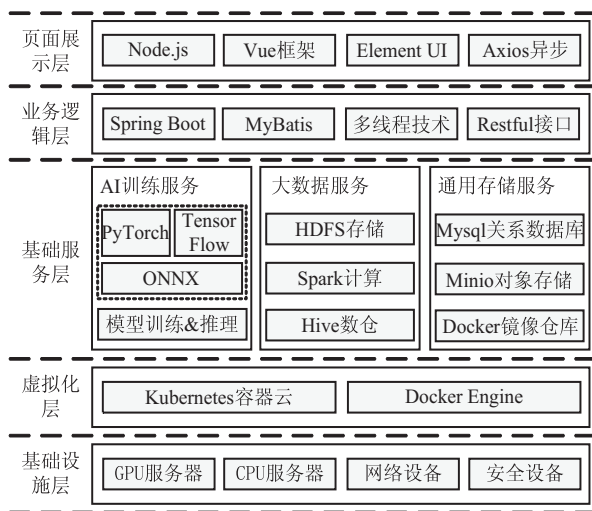


图 11 平台技术架构

5 结束语

传统企业的数据智能业务发展对于当前过载的互联网、大数据环境显得力不从心。针对海量多源异构数据集中管理与利用的问题、服务开发过程依旧繁琐、开发周期依旧长的问题和服务之间的可重复利用问题, 设计并实现了一个云原生数据智能服务开发平台, 提高了从数据资产到模型资产持续训练和集成的价值转变能力, 模型资产到服务资产持续部署的自动化能力, 以及服务资产的一站式运维能力和服务资产之间的可重复利用能力, 使相关人员能够通过该平台快速完成数据智能服务的开发与管理。研发的平台支撑了国家自然科学基金医学认知图谱复杂知识结构表示研究项目。

参考文献:

- [1] 时晓晖, 时晓虹, 吴 雷. 企业数字化转型面临的困境与对策研究[J]. 投资与创业, 2022, 33(14): 163-166.
- [2] BISONG E. Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners[M]. Berkeley: Apress, 2019: 671-685.
- [3] HUMMER W, MUTHUSAMY V, RAUSCH T, et al. Modelops:

(下转第 85 页)