(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 112463141 B
(45)授权公告日 2021.12.14

(21)申请号 202011338284.7

(22)申请日 2020.11.25

(65)同一申请的已公布的文献号
申请公布号 CN 112463141 A

(43)申请公布日 2021.03.09

(73)专利权人 北京航空航天大学
地址 100191 北京市海淀区学院路37号

(72)发明人 吴文峻 于笑明 王洋洲

(74)专利代理机构 北京永创新实专利事务所 11121
代理人 易卜

(51)Int.Cl.
*G06F 8/34*(2018.01)
*G06F 8/38*(2018.01)
*G06F 8/30*(2018.01)

(56)对比文件
CN 111966744 A,2020.11.20
CN 110908651 A,2020.03.24
CN 110764752 A,2020.02.07
CN 101821709 A,2010.09.01
US 2006229925 A1,2006.10.12
US 2019332357 A1,2019.10.31
CN 108681450 A,2018.10.19
CN 110880091 A,2020.03.13
CN 101695080 A,2010.04.14
Giuseppe Tricomi等.Orchestrated Multi-Cloud Application Deployment in OpenStack with TOSCA.《https://ieeexplore.ieee.org/abstract/document/7947027》.2017,第1-6页.
王刚.基于BPMN规范的工作流引擎设计与实现.《计中国优秀硕士学位论文全文数据库 信息科技辑》.2020,第I138-108页.
Jianhong Ye等.Transformation of BPMN Diagrams to YAWL Nets.《Journal of Software》.2010,第1-6页.

审查员 吴银娥

权利要求书2页 说明书7页 附图4页

(54)发明名称

一种基于BPMN的微服务工作流部署方法

(57)摘要

本发明公开了一种基于BPMN的微服务工作流部署方法,涉及工作流引擎领域,具体为:用户通过流程编辑界面串接各图元,以bpmn格式进行存储,然后筛选各个图元的每个元素,将其分为节点类元素和关系类元素;对每个节点类中各元素,去除变量名id和值,将剩余的变量名和值写入YAML中的数组对象template中;同时,对每个关系类中各元素,根据变量名start和end中的变量值,得到工作流的执行顺序,并写入YAML中的数组对象dependencies中;将添加完声明信息的YAML文件提交给工作流部署引擎,针对当前微服务,引擎读取对象类DAG,获得元素的先后执行顺序;并查找对象类Templates中对应的元素,按顺序将所有元素执行完毕后,对节点类的微服务执行完毕;本发明大大降低了微服务部署的门槛。




CN 112463141 B

1.一种基于BPMN的微服务工作流部署方法,其特征在于,所述步骤如下:

步骤一、用户通过流程编辑界面设计一个仅包括图元和自然语言描述的工作流程,以BPMN格式进行存储;

用户根据需求有序地将各图元进行串接,同时用自然语言描述编辑后保存,得到BPMN后缀的文件;

步骤二、将读取的BPMN信息分为两类,并筛选出各个图元及其属性的信息;

两类包括:BPMN各个图元及其属性,以及BPMN中各个图元在画布上的形状与位置信息;

步骤三,针对筛选出的各图元,根据属性信息对图元中的每个元素进行分类;

将各图元中属性信息为活动的元素归为一类,简称流程的节点类;将各图元中属性的信息为箭头或条件的元素划为一类,简称流程的关系类;

步骤四、针对每个节点类中各元素,将当前元素的属性内容中按变量名和变量值进行遍历,去除变量名id以及对应的变量值,将剩余的变量名写入YAML中的数组对象template中;

每个节点类对应一个微服务,所有的微服务均存储在仓库中;

具体为:针对当前节点类A中选取元素B,对元素B中的各变量名进行遍历,去除变量名id以及对应的变量值,将剩余的变量名和变量值作为数组对象,映射写入YAML中的数组对象template中;同时,在仓库中查询每个微服务,将微服务对应的name信息作为前缀名字,存入数组对象template中dag->tasks的name字段中;

元素B中信息存入数组对象template后,按BPMN中的顺序依次选择下一个元素C,将元素C中信息遍历完毕,写入YAML中对应的数组对象template中;

当前节点类A中所有元素的变量值以缩进方式作为属性存入,当前节点类的元素都遍历完毕后,选择下一个节点类作为当前节点类,继续以同样的方式遍历,最终所有节点类的存储形成对象类Templates;

步骤五、针对每个关系类中各元素,根据各元素的变量名start和end中的变量值,得到工作流的执行顺序,并写入YAML中的数组对象dependencies中,所有关系类元素的数组对象dependencies组成对象类DAG;

具体为:

每个关系类的元素中,变量名start和end中存储的值分别为元素,针对关系类元素r的变量名start的变量值为元素“A”,变量名end的变量值为元素“B”,则元素r的描述关系为先执行元素A后执行元素B,则将变量值“A”存入元素B的数组对象dependencies中;

所有的关系类元素变量名的数组对象dependencies,共同组成对象类DAG;

步骤六、对当前BPMN文件读取完后,在YAML中的数组对象template和dependencies对应的对象类Templates和DAG上,根据用户需求添加声明信息;

步骤七、将添加完声明信息的YAML文件提交给工作流部署引擎,引擎读取对象类DAG,根据对象类DAG中各元素的dependencies属性获得工作流中各元素的先后执行顺序;并查找Templates中对应的元素,按顺序将所有元素执行完毕后,对节点类的微服务执行完毕;

步骤八、当各节点类的微服务全部执行完毕后,用户设计的工作流程部署完毕。

2.如权利要求1所述的一种基于BPMN的微服务工作流部署方法,其特征在于,所述的步骤一中,流程编辑界面支持用户使用拖拽方式绘制各类图元,其中图元指的是一组图形元

素;图形元素分为四种:流对象、连接对象、甬道和描述对象。

3.如权利要求1所述的一种基于BPMN的微服务工作流部署方法,其特征在于,所述的步骤四中,每个元素分别对应一个数组对象template和一个task块。

4.如权利要求1所述的一种基于BPMN的微服务工作流部署方法,其特征在于,所述的步骤五中,针对两个需同步执行的关系类元素,具体处理如下:

元素N的执行同时依赖元素P和元素Q;元素P和元素N之间的关系类元素为r,元素r的变量名start的变量值为元素“P”,变量名end的值为“N”,元素Q和元素N之间的关系类元素为s,元素s的变量名start的值为“Q”,变量名end的值为“N”;

则描述关系为先执行P和Q后执行N,元素P和Q同步,或者根据用户需求先后顺序执行,将值“P”和“Q”分别存入元素N的数组对象dependencies中。

5.如权利要求1所述的一种基于BPMN的微服务工作流部署方法,其特征在于,所述的步骤七具体为:用户设计的工作流,每一个步骤分别对应一个容器;

首先对象类Templates中定义了各节点类的微服务,工作流引擎根据当前微服务的内容创建新的容器,每个容器通过添加sidecar解析容器中的文件运行具体的微服务;

工作流引擎读取对象类DAG中各元素的dependencies属性获得工作流中各元素的先后执行顺序;在Templates中查找所涉及的微服务全部完成的元素;按顺序将所有元素执行完毕后,当前微服务执行完毕,将状态变成已完成状态;

工作流引擎继续选择下一个微服务,并创建新的容器,利用sidecar解析下一个微服务,读取对象类DAG中元素的dependencies属性,查找下一个微服务需要的全部元素,即为下一个微服务进行部署。

# 一种基于BPMN的微服务工作流部署方法

## 技术领域

[0001] 本发明涉及微服务工作流引擎领域，尤其涉及一种基于BPMN的微服务工作流部署方法。

## 背景技术

[0002] 随着互联网业务的不断发展，随之而来的用户量增长、系统数量增多以及调用依赖关系更加复杂化，系统架构从单体时代迁移到了服务SOA时代。微服务即是SOA下的产物，这种架构提倡将应用程序划分成一组小的服务，每个服务独立地运行在自己的进程中，服务之间采用轻量级的通信机制互相沟通，每个服务都围绕着具体的业务进行构建。

[0003] 随着微服务的流行与发展，容器技术也相应得被重视起来。容器包含了完整的运行时环境:除了应用程序本身之外，这个应用所需的全部依赖、类库、其他二进制文件和配置文件等，都统一被打入了一个称为容器镜像的包中。通过将应用程序本身和其依赖容器化，抽象化了操作系统发行版本和其他基础环境造成的差异。起初人们倾向于将微服务部署在虚拟机里，这能够实现微服务的隔离性，但是无法进行快速扩展，无法立即对瞬时突增的负载或者流量做出反应。

[0004] 虚拟机是在物理服务器的上层，用软件来模拟特定的硬件系统。Hypervisor位于硬件和系统之间，是创建虚拟机必须的一个部分。虚拟机软件必须使用Hypervisor作为一个中间层，是虚拟机技术的核心，当宿主操作系统启动虚拟机时，会通过hypervisor给虚拟机分配内存，CPU，网络和磁盘等资源，并加载虚拟的操作系统，因而需要消耗宿主机大量的物理资源。而使用了容器技术，一台宿主机上运行的多个容器化应用共享这台宿主机操作系统的内核，因而不需要虚拟机技术的hypervisor中间层，同虚拟机技术相比，更加轻量化，启动速度更快。

[0005] 从成本考虑，使用传统的虚拟机技术，为了实现隔离性，每个应用或者说微服务都必须运行在一个虚拟机里，这种重复和浪费的操作系统和资源的分配，可以通过容器技术来避免，从而大大减少了云服务提供商对硬件的投入，节省了云服务中心的成本。

[0006] 现在已经有不少基于容器技术建立的微服务平台，如Kubernetes，但是用户在使用Kubernetes部署微服务的时候，需要学习如何编写部署的YAML文档，其中包括Kubernetes中对各种资源的定义与实现、部署文档的格式要求等等知识，用户上手使用需要付出一定的学习成本。虽然Kubernetes能够支持微服务的部署，但是其本身并没有带有调用运行和管理微服务的流程的功能。

[0007] BPMN是BPM及workflow的建模语言标准之一。BPMN定义了一个业务流程图(Business Process Diagram)，该业务流程图基于一个流程图(flowcharting)，该流程图被设计用于创建业务流程操作的图形化模型。业务流程图由一系列的图形化元素组成。这些元素简化了模型的开发，且业务人员看上去非常熟悉。这些元素每个都有各自的特性，且与大多数的建模器类似。比如，活动是矩形，条件是菱形。通过将BPMN转换为YAML来部署工作流程，业务人员操作更直观、更便捷。

[0008] BPMN是基于XML语言书写的,Kubernetes的资源定义则都是基于YAML语言的规范,想要将BPMN的可视化流程建模应用于Kubernetes上,就必须对两种语言进行转化。而现有的算法都是XML语言直接转换成YAML,XML中以标签形式体现的元素在YAML中转换为缩进的体现,没有专门针对BPMN规范的转换。并且现有的技术并没有对BPMN中的信息进行分析筛选统合,只是单纯的格式转换,这样得到的YAML文件不仅包含了不少多余的信息同时也无法满足部署系统的需求。

## 发明内容

[0009] 本发明针对上述问题,提供了一种基于BPMN的微服务工作流部署方法,将实际业务工作流使用微服务的架构部署到容器中,不需要用户学习如何编写部署文档,即可实现设计工作流到实际部署完成工作流程的全过程,大大降低了微服务部署的门槛。

[0010] 所述的微服务工作流部署方法,分以下步骤:

[0011] 步骤一、用户通过流程编辑界面设计一个仅包括图元和自然语言描述的工作流程,以bpmn格式进行存储;

[0012] 流程编辑界面支持用户使用拖拽方式绘制各类图元,其中图元指的是一组图形元素;

[0013] 用户根据需求有序地将各图元进行串接,同时用自然语言描述编辑后保存,得到bpmn后缀的文件;

[0014] 图形元素分为四种:流对象、连接对象、甬道和描述对象。

[0015] 步骤二、将读取的bpmn信息分为两类,并筛选出各个图元及其属性的信息;

[0016] 两类包括:bpmn各个图元及其属性,以及bpmn中各个图元在画布上的形状与位置信息。

[0017] 然后,筛选出bpmn中各个图元及其属性的信息,并将bpmn中各个图元在画布上的形状与位置信息舍去。

[0018] 步骤三,针对筛选出的各图元,根据属性信息对图元中的每个元素进行分类;

[0019] 将各图元中属性信息为活动的元素归为一类,简称流程的节点类;将各图元中属性的信息为箭头或条件的元素划为一类,简称流程的关系类;

[0020] 步骤四、针对每个节点类中各元素,将当前元素的属性内容中按变量名和变量值进行遍历,去除变量名id以及对应的变量值,将剩余的变量名写入YAML中的数组对象template中。

[0021] 每个节点类对应一个微服务,所有的微服务均存储在仓库中;

[0022] 具体为:针对当前节点类A中选取元素B,对元素B中的各变量名进行遍历,去除变量名id以及对应的变量值,将剩余的变量名和变量值作为数组对象,映射写入YAML中的数组对象template中。同时,在仓库中查询每个微服务,将微服务对应的name信息作为前缀名字,存入数组对象template中dag->tasks的name字段中。

[0023] 每个元素分别对应一个数组template和一个task块;

[0024] 元素B中信息存入数组template后,按bpmn中的顺序依次选择下一个元素C,将元素C中信息遍历完毕,写入YAML中对应的数组对象template中。

[0025] 当前节点类A中所有元素的变量值以缩进方式作为属性存入,当前节点类的元素

都遍历完毕后,选择下一个节点类作为当前节点类,继续以同样的方式遍历,最终所有节点类的存储形成对象类Templates;

[0026] 步骤五、针对每个关系类中各元素,根据各元素的变量名start和end中的变量值,得到工作流的执行顺序,并写入YAML中的数组对象dependencies中,所有关系类元素的数组对象dependencies组成对象类DAG。

[0027] 具体为:

[0028] 每个关系类的元素中,变量名start和end中存储的值分别为元素,针对关系类元素r的变量名start的变量值为元素“A”,变量名end的变量值为元素“B”,则元素r的描述关系为先执行元素A后执行元素B,则将变量值“A”存入元素B的数组对象dependencies中;

[0029] 所有的关系类元素变量名的数组对象dependencies,共同组成对象类DAG的一项数组;

[0030] 步骤六、对当前bpmn文件读取完后,在YAML中的数组对象template和dependencies对应的对象类Templates和DAG上,根据用户需求添加声明信息;

[0031] 声明信息比如:命名空间。

[0032] 步骤七、将添加完声明信息的YAML文件提交给工作流部署引擎,引擎读取对象类DAG,根据对象类DAG中各元素的dependencies属性获得工作流中各元素的先后执行顺序;并查找Templates中对应的元素,按顺序将所有元素执行完毕后,对节点类的微服务执行完毕。

[0033] 具体为:用户设计的工作流,每一个步骤分别对应一个容器,

[0034] 首先对象类Templates中定义了各节点类的微服务,工作流引擎根据当前微服务的内容创建新的容器,每个容器通过添加sidecar解析容器中的文件运行具体的微服务;

[0035] 工作流引擎读取对象类DAG中各元素的dependencies属性获得工作流中各元素的先后执行顺序;在Templates中查找所涉及的微服务全部完成的元素;按顺序将所有元素执行完毕后,当前微服务执行完毕,将状态变成已完成状态;

[0036] 工作流引擎继续选择下一个微服务,并创建新的容器,利用sidecar解析下一个微服务,读取对象类DAG中元素的dependencies属性,查找下一个微服务需要的全部元素,即为下一个微服务进行部署。

[0037] 步骤八、当各节点类的微服务全部执行完毕后,用户设计的工作流程部署完毕。

[0038] 本发明的优点在于:

[0039] 本发明一种基于BPMN的微服务工作流部署方法,利用可视化的流程编辑界面,对用户友好,用户只需要与可视化界面进行交互,无需关心部署的细节只需将精力集中于工作流的创建上,考虑微服务之间的关系,节省了用户在使用上的学习成本,简化了用户的操作。

## 附图说明

[0040] 图1为本发明一种基于BPMN的微服务工作流部署方法的流程图;

[0041] 图2为本发明采用的基于BPMN的工作流程编辑创建的前端界面;

[0042] 图3为本发明实施例建立的基于BPMN的微服务工作流部署的工作模块;

[0043] 图4为本发明实施例采用的部署方案生成方法流程图;

[0044] 图5为本发明实施例采用的物理部署方法流程图。

## 具体实施方式

[0045] 下面将结合附图和实施实例对本发明作进一步的详细说明。

[0046] 本发明一种基于BPMN的微服务工作流部署方法,如图1所示,分以下步骤:

[0047] 步骤一、用户通过流程编辑界面设计一个仅包括图元和自然语言描述的工作流程,以bpmn格式进行存储;

[0048] 如图2所示,流程编辑界面支持用户使用拖拽方式绘制各类图元,其中图元指的是一组图形元素;比如活动是长方形,路由是菱形。界面左侧是一系列符合BPMN2.0规范的图形化元素,中间为画布,用户在画布上编辑流程,上方按钮为支持的一些功能,一键部署功能会将用户完成的流程文件发送给BPMN转YAML的组件。右侧显示该界面也可支持用户对每个图形化元素进行一些定义,id为拖曳到画布中自动为该元素生成的随机id。

[0049] 界面功能包括:流程图元的列表;流程图元的编辑、拖拽、增加、删除;流程图元属性值的修改;

[0050] 用户根据需求有序地将各图元串接为完整的业务流程,同时用自然语言描述编辑后保存,得到bpmn后缀的文件;

[0051] 图形元素分为四种:流对象(Flow Objects)、连接对象(Connecting Objects)、甬道(Swimlanes)和描述对象(Artifacts)。

[0052] 步骤二、将读取的bpmn信息分为两类,并筛选出各个图元及其属性的信息;

[0053] 两类包括:bpmn各个图元及其属性,以及bpmn中各个图元在画布上的形状与位置信息。

[0054] 然后,根据各元素的变量名筛选出bpmn中各个图元及其属性的信息,并将bpmn中各个图元在画布上的形状与位置信息舍去。

[0055] 步骤三,针对筛选出的各图元,根据属性信息对图元中的每个元素进行分类;解析出工作流中每一步所需的微服务与微服务的依赖关系;

[0056] 基于bpmn2.0的规范,对每个图形化元素进行分类,根据XML语言中定义的元素变量名来分析bpmn,将各图元中属性信息为活动的元素归为一类,简称流程的节点类;将各图元中属性的信息为箭头或条件的元素划为一类,简称流程的关系类;即将其分为流程的节点和流程的节点关系两类。

[0057] 每个流程节点类对应一项具体的事务,该事务映射到Kubernetes中的一个微服务,所有的微服务均存储在仓库中;而这些节点关系就是微服务之间的依赖关系。

[0058] 步骤四、针对每个节点类中各元素,将当前元素的变量名和变量值进行遍历,去除变量名id以及对应的变量值,将剩余的变量名写入YAML中的数组对象template中,通过在数组对象template中添加微服务对应的name,形成对象类Templates;

[0059] 每个元素包括若干不同的变量名和对应的变量值;元素个数不定,根据用户的需求设定;

[0060] 具体为:对当前节点类A中各元素进行遍历,将当前元素中变量名id以及对应的变量值去除,将剩余的变量名和变量值作为数组对象,映射写入YAML中的数组对象template中。

[0061] 同时，在仓库中查询每个流程节点对应的微服务，将微服务对应的name信息作为前缀名字，存入YAML中当前元素对应的数组template中。

[0062] template是一个数组对象，存储的是每个元素除去变量名id后，剩余的变量名和变量值，以及微服务对应的name信息；每个元素分别对应一个数组template；

[0063] 元素A中的当前元素存入数组template后，按bpmn中的顺序依次选择下一个元素进行遍历，直至所有元素都遍历完毕；并一一写入YAML中各元素对应的数组对象template中。

[0064] 当前节点类A中每个元素以缩进方式作为属性存入，以数组形式存储到template中，直至所有内容都遍历完毕，选择下一个节点类作为当前节点类，将各元素重复遍历，最终所有的节点类形成对象类Templates。

[0065] 步骤五、针对每个关系类中各元素，根据各元素的变量名start和end中的变量值，得到工作流的执行顺序，并写入YAML中的数组对象dependencies中，所有关系类元素的数组对象dependencies组成对象类DAG。

[0066] 流程的节点关系是事务执行的先后顺序，因为存在某些事务需要在其他事务完成的基础上才能执行。在bpmn中以有向线段的图形表示，该元素描述了有向线段的两端节点，例如：关系类元素r的变量名start的值为元素“A”，变量名end的值为元素“B”，则描述关系为先执行元素A后执行元素B，将变量值A存入元素B的数组对象dependencies中，形成对象类DAG的一项元素。

[0067] 同理，将所有关系类中各元素，根据变量名将对应的变量值存储到数组对象dependencies中，共同组成对象类DAG，用于描述流程关系。

[0068] 在有些流程中，某个元素N的执行可能同时依赖两个元素，如元素P和元素Q；

[0069] 此时，元素P和元素N之间的关系类元素为r，元素r的变量名start的变量值为元素“P”，变量名end的值为“N”，元素Q和元素N之间的关系类元素为s，元素s的变量名start的值为“Q”，变量名end的值为“N”；

[0070] 则描述关系为先执行P和Q后执行N，元素P和Q可以同步，或者根据用户需求先后顺序执行，将值“P”和“Q”分别存入元素N的数组对象dependencies中。

[0071] 步骤六、对当前bpmn文件读取完后，在YAML中的数组对象template和dependencies对应的对象类Templates和DAG上，根据用户需求添加必要的声明信息；

[0072] 如namespace声明该流程需要在集群中的哪个命名空间里进行实现，并且按照YAML的语言规范要求进行生成文件，规范要求包括缩进、空格等。

[0073] 步骤七、将添加完声明信息的YAML文件提交给工作流部署引擎，引擎读取对象类DAG，根据对象类DAG中各dependencies的属性获得各元素的先后执行顺序，并查找Templates中对应的元素，按各元素按顺序执行完毕后，当前节点类的微服务执行完毕。

[0074] 该引擎定义了Kubernetes的一种新的资源用于表示工作流，在工作流的每一个步骤都对应着一个容器。该引擎首先会读取对象类DAG，根据DAG中dependencies属性获得各元素对应流程的先后执行顺序，如果dependencies属性中没有值的即为流程中第一个执行；根据对象类DAG中存储的name，查找数组对象template中对应的元素。

[0075] 对象类Templates中定义了该节点对应的微服务的具体信息，包括名称、镜像、端口等，工作流引擎会依据这些信息在Kubernetes上创建新的容器，容器中运行具体的微服

务。待该服务变成已完成状态时，工作流引擎便继续读取对象类DAG中元素的dependencies属性，查找dependencies中所涉及的服务全部完成的元素，即为下一个进行部署的服务。

[0076] 在工作流的每一个步骤都对应着一个容器，每一个容器添加了一个sidecar用于解析容器中的文件。Sidecar技术Kubernetes本身已经实现，它可以与主容器进行交互，依赖这个技术将容器中的文件进行解析，并存储到临时数据库中。在建立下一步容器时，会同时创建一个新的sidecar，将下一步容器所需要用到的文件解析出来放到下一步的容器中。它从临时数据库中解析文件，并交给主容器，由此实现流程中每个步骤的信息传递。

[0077] 步骤八、当各节点类的微服务全部执行完毕后，用户设计的工作流程部署完毕。

[0078] 本发明一方面，提供可视化的流程编辑界面，支持使用拖拽方式绘制流程的各类图元，有序地串接为完整的业务流程；提供对流程属性、流程中各业务环节属性的即时设置，以及流程的保存、提交和部署。

[0079] 另一方面，本发明对BPMN工作流文件进行解析；提取业务流程中的每一个步骤的信息并匹配得到需要部署的微服务；解析工作流中每个步骤之间的依赖关系；生成物理部署方案。根据部署方案在仓库中寻找需要部署的微服务；将多步骤工作流建模为一系列任务，使用图(DAG)建立任务之间的依赖关系；根据依赖关系将微服务依次部署到工作流引擎上。

[0080] 实施例：

[0081] 如图3所示，本实施基于流程创建模块、部署方案生成模块与物理部署模块进行操作：

[0082] 所述流程创建模块提供给用户可视化界面进行工作流程编辑；

[0083] 所述部署方案生成模块，用于解析用户创建的工作流，生成微服务工作流的部署方案；

[0084] 所述物理部署模块，用于解析部署方案，运用容器技术实际部署微服务。

[0085] 本发明实施例部署方案生成的流程图，如图3所示，包括：

[0086] 301、接收流程创建模块发送的流程文件；

[0087] 302、根据流程文件中携带的信息，提取用户对工作流中每一步的具体描述，分析出用户所需要部署的微服务；

[0088] 303、根据流程文件中携带的信息，分析工作流中每一步之间的依赖关系，得到微服务之间的依赖关系；

[0089] 304、整理微服务及他们之间的依赖关系，按照相应的格式生成部署文件后发送给物理部署模块。

[0090] 流程文件的解析方法包括：读取流程文件，分辨出文件中所描述的图形信息和用户所设置的工作流环节属性信息，根据用户所设置的工作流环节属性信息的描述通过对比知识图谱分析得到用户需要部署的微服务，再将微服务与流程文件的图元一一对应，根据图形信息所提供的图元之间的关系得到微服务之间的依赖关系。

[0091] 本发明实施例物理部署方法，如图5所示，包括：

[0092] 401、接收部署方案生成模块发送的部署文件；

[0093] 402、根据部署文件中携带的信息，提取需要部署的微服务；

[0094] 403、根据部署文件中携带的信息，提取微服务之间的依赖关系，使用图(DAG)建立

任务之间的依赖关系；

[0095] 404、根据微服务的依赖关系，依次部署微服务。

[0096] 根据微服务的依赖关系部署微服务的方法包括：

[0097] 服务B需要在服务A的基础上进行部署，先部署服务A再部署服务B；服务B的运行需要服务A执行的结果，先部署服务A，为服务A多对应容器a添加sidecar容器解析需要的文件和信息，将解析到的文件先放置在临时中转站中，再部署服务B，将中转站中的文件发送给服务B所对应的容器b，进而完成这个过程。


用户通过流程编辑界面设计一个仅包括图元和自然语言描述的工作流程，以bpmn格式进行存储
将读取的bpmn信息分为两类，并筛选出各个图元及其属性的信息
针对筛选出的各图元，根据属性信息对图元中的每个元素进行分类
针对每个节点类中各元素，将当前元素的属性内容中按变量名和变量值进行遍历，去除变量名id以及对应的变量值，将剩余的变量名写入YAML中的数组对象template中
针对每个关系类中各元素，根据各元素的变量名start和end中的变量值，得到工作流的执行顺序，并写入YAML中的数组对象dependencies中，所有关系类元素的数组对象dependencies组成对象类DAG
对当前bpmn文件读取完后，在YAML中的数组对象template和dependencies对应的对象类Templates和DAG上，根据用户需求添加声明信息
将添加完声明信息的YAML文件提交给工作流部署引擎，引擎读取对象类DAG，根据对象类DAG中各元素的dependencies属性获得工作流中各元素的先后执行顺序；并查找Templates中对应的元素，按顺序将所有元素执行完毕后，对节点类的微服务执行完毕
当各节点类的微服务全部执行完毕后，用户设计的工作流程部署完毕

图1

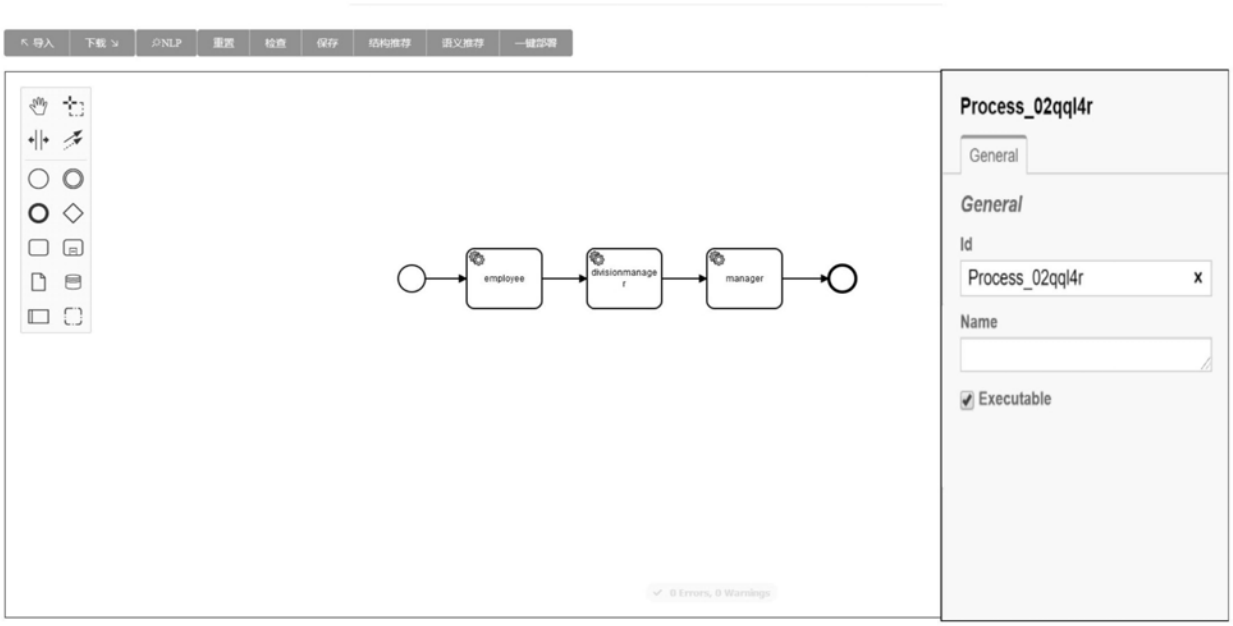

导入
下载
NLP
重置
检查
保存
结构推荐
语义推荐
一键部署
employee
divisionmanager
manager
Process_02qql4r
General
General
Id
Process_02qql4r
Name
Executable

图2

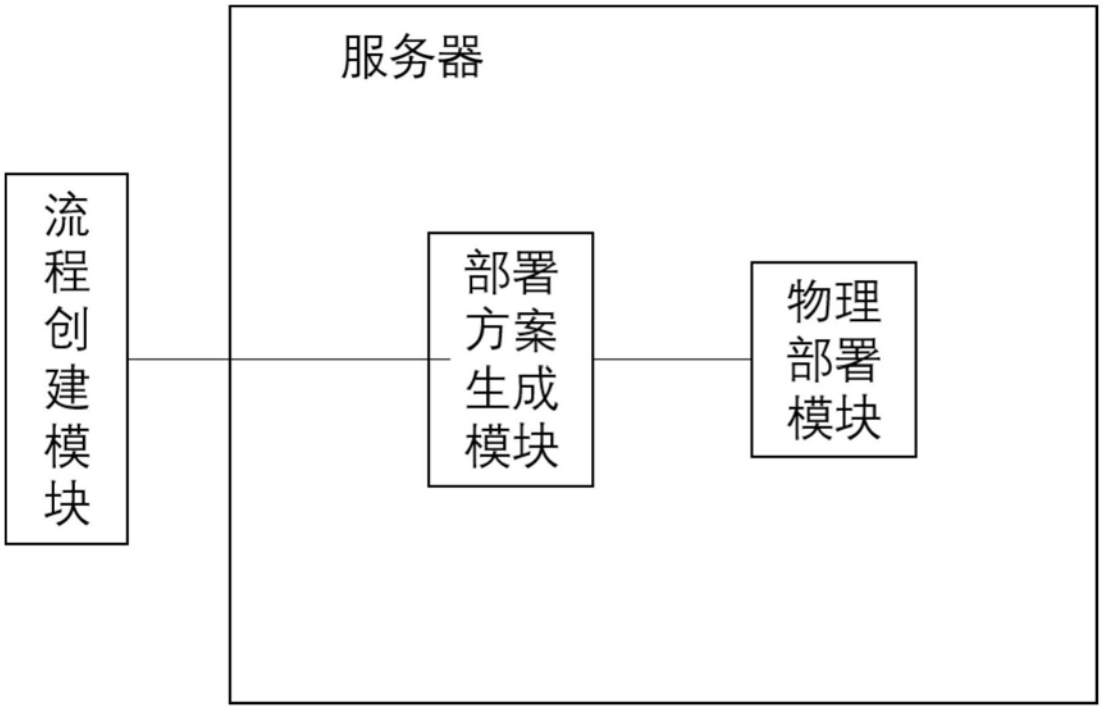

服务器
流程创建模块
部署方案生成模块
物理部署模块

图3

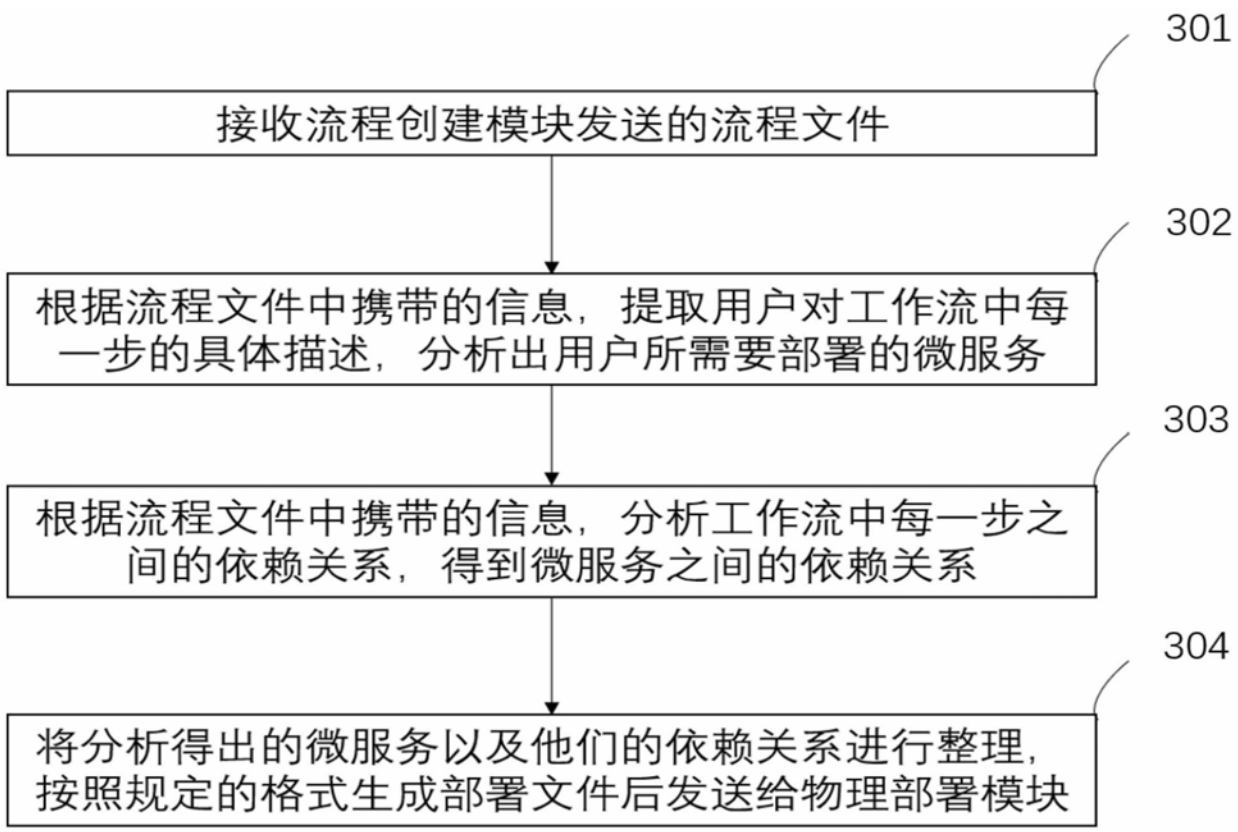

301
接收流程创建模块发送的流程文件
302
根据流程文件中携带的信息，提取用户对工作流中每一步的具体描述，分析出用户所需要部署的微服务
303
根据流程文件中携带的信息，分析工作流中每一步之间的依赖关系，得到微服务之间的依赖关系
304
将分析得出的微服务以及他们的依赖关系进行整理，按照规定的格式生成部署文件后发送给物理部署模块

图4

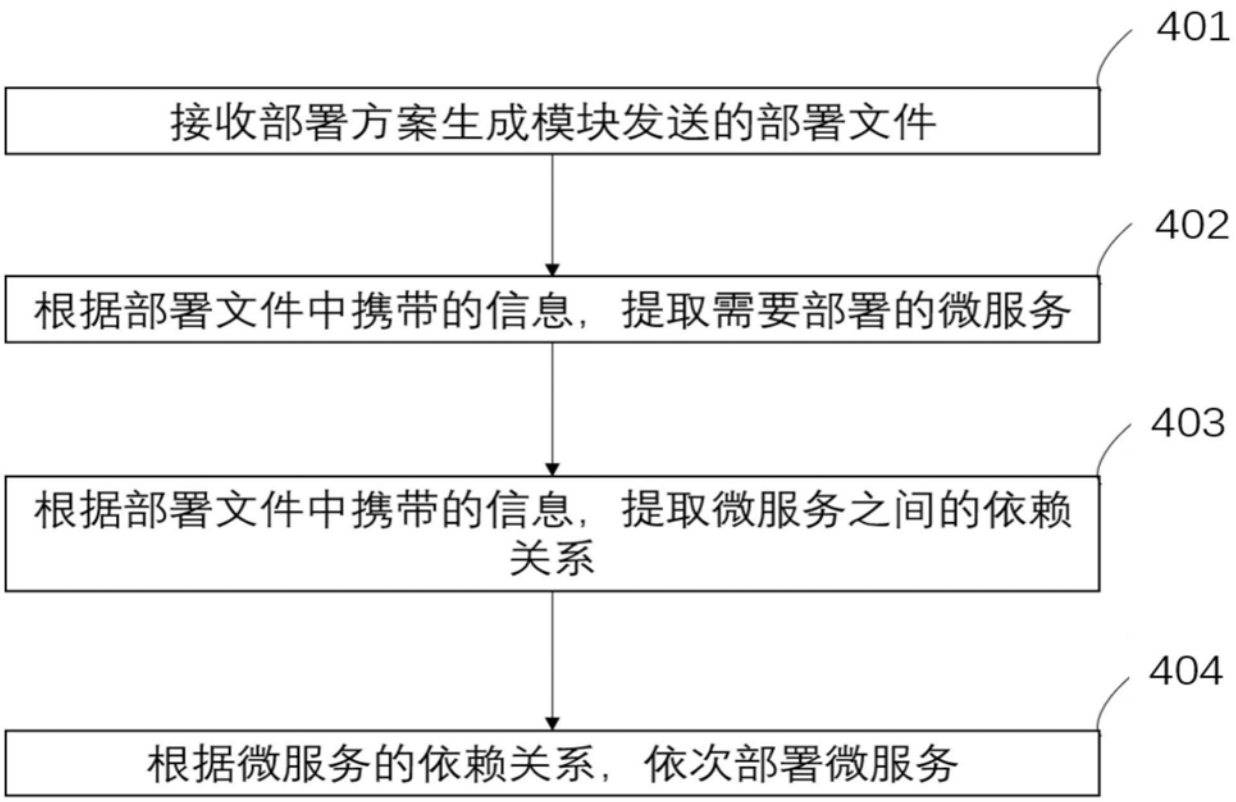

401
接收部署方案生成模块发送的部署文件
402
根据部署文件中携带的信息，提取需要部署的微服务
403
根据部署文件中携带的信息，提取微服务之间的依赖关系
404
根据微服务的依赖关系，依次部署微服务

图5