



(12) 发明专利申请

(10) 申请公布号 CN 119166325 A

(43) 申请公布日 2024.12.20

(21) 申请号 202410943266.3

(22) 申请日 2024.07.15

(71) 申请人 抚顺市亿腾网络科技有限公司

地址 113000 辽宁省抚顺市望花区朝阳路
东段43号楼3单元502号

(72) 发明人 请求不公布姓名

(51) Int.Cl.

G06F 9/50 (2006.01)

G06F 9/455 (2018.01)

G06F 9/54 (2006.01)

权利要求书2页 说明书5页 附图1页

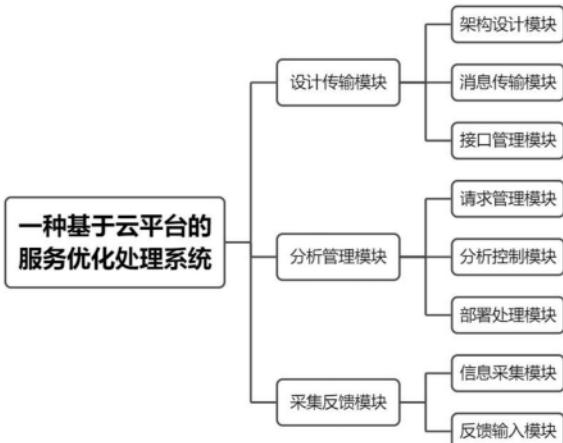
(54) 发明名称

一种基于云平台的服务优化处理系统

(57) 摘要

本发明公开了一种基于云平台的服务优化处理系统，该系统的运行方法包括：进行云平台服务系统的架构设计管理；对云平台服务系统进行消息传输及接口管理；进一步对云平台服务系统进行分析控制；进行云平台服务系统的采集反馈管理；所述进行云平台服务系统的架构设计管理包括：在OpenStack云平台服务系统中，使Sdocke通过调用OpenStack中Keystone、Glance以及Neutron组件暴露的API接口获取创建容器所需的镜像和网络资源，并通过调用Docker提供的API接口创建容器以及对容器进行更新处理。所述对云平台服务系统进行消息传输及接口管理包括：控制云平台服务系统组件间通过消息队列进行传输通信。本发明，具有智能管理控制及处理效率高的特点。

A
CN 119166325



1.一种基于云平台的服务优化处理方法,其特征在于:包括:

进行云平台服务系统的架构设计管理;

对云平台服务系统进行消息传输及接口管理;

进一步对云平台服务系统进行分析控制;

进行云平台服务系统的采集反馈管理;

所述进行云平台服务系统的架构设计管理包括:

在OpenStack云平台服务系统中,使Sdocker通过调用OpenStack中Keystone、Glance以及Neutron组件暴露的API接口获取创建容器所需的镜像和网络资源,并通过调用Docker提供的API接口创建容器以及对容器进行更新处理。

2.根据权利要求1所述的一种基于云平台的服务优化处理方法,其特征在于:所述对云平台服务系统进行消息传输及接口管理包括:

控制云平台服务系统组件间通过消息队列进行传输通信,包括:生产者(Producer)、交换器(Exchange)、绑定(Binding)、队列(Queue)和消费者(Consumer)。

3.根据权利要求1所述的一种基于云平台的服务优化处理方法,其特征在于:所述对云平台服务系统进行消息传输及接口管理进一步包括:

通过Sdocker的sdock_API为用户提供RESTful API,使用户可以通过HTTP请求的方式方便快捷地查看容器及OpenStack中各个计算节点的相关信息。

4.根据权利要求1所述的一种基于云平台的服务优化处理方法,其特征在于:所述进一步对云平台服务系统进行分析控制包括:

在云平台服务系统中,当用户开启Sdocker的负载均衡调度模式并指定容器资源信息(CPU、内存、磁盘三种资源信息),向Sdocker请求创建容器,用户请求创建容器的命令里包含容器名称、容器镜像、容器网络信息以及容器的资源请求规格信息;

接收到容器调度请求信息后,sdock_Scheduler组件首先对容器的资源信息进行解析,如果容器具有CPU、内存和磁盘信息,则对容器实施调度,在进行容器调度时均采用LBSM调度机制为容器选取最优计算节点和最优NUMA节点;

sdock_Compute获取到的image信息后,根据容器的名称和容器资源请求信息,调用Docker提供的API接口创建并启动容器,Sdock_Compute依托解析出的token,调用Neutron组件暴露的API接口创建port,并通过port在本地计算节点创建虚拟网卡,将该网卡挂载至该容器的网络命名空间,完成容器部署。

5.根据权利要求1所述的一种基于云平台的服务优化处理方法,其特征在于:所述进行云平台服务系统的采集反馈管理包括:

对云平台服务系统的运行数据信息根据设定的周期进行采集,并将其传输至工作台处,使工作人员可以进行对应的监测分析;

用户可对云服务的使用进行意见反馈输入,为后续云平台服务系统维护优化提供有效的参考依据。

6.一种基于云平台的服务优化处理系统,其特征在于:包括:

设计传输模块,用于进行云平台服务系统的设计传输控制;

分析管理模块,用于进行云平台服务系统的分析优化管理;

采集反馈模块,用于进行云平台服务系统的信息采集反馈。

7. 根据权利要求6所述的一种基于云平台的服务优化处理系统,其特征在于:所述设计传输模块包括:

架构设计模块,用于对云平台服务系统进行架构设计管理;
消息传输模块,用于进行云平台服务系统组件间消息传输处理;
接口管理模块,用于进行云平台服务系统接口控制。

8. 根据权利要求6所述的一种基于云平台的服务优化处理系统,其特征在于:所述分析管理模块包括:

请求管理模块,用于进行云平台服务系统的负载均衡请求管理;
分析控制模块,用于进行云平台容器调度分析控制;
部署处理模块,用于进行云平台容器部署分析处理。

9. 根据权利要求6所述的一种基于云平台的服务优化处理系统,其特征在于:所述采集反馈模块包括:

信息采集模块,用于进行云平台服务系统的信息采集;
反馈输入模块,用于进行数据信息的反馈输入管理。

一种基于云平台的服务优化处理系统

技术领域

[0001] 本发明涉及云平台技术领域,具体为一种基于云平台的服务优化处理系统。

背景技术

[0002] 云平台作为云计算领域的资源管理平台,其高可用性可以保证用户服务质量,高灵活性可以实现资源的统一调配与弹性伸缩。同时,云平台提供的多种调用方式以及丰富的API接口,为用户进行其他应用的开发提供了极大便利。Docker容器技术具有快速部署与交付,灵活迁移与扩展以及高资源利用率等优势,OpenStack云平台具有开源、兼容、灵活和可扩展等优势,二者的融合可以为用户提供CaaS解决方案,为Docker容器提供高效灵活的管控,为OpenStack带来更优异的性能表现和更高的社区活跃度。但现有的容器调度方案多基于用户对容器的初始资源请求对容器进行调度并分配资源,未充分考虑到容器运行时的实际资源使用情况,从而严重影响云平台的运行处理。因此,设计智能管理控制及处理效率高的一种基于云平台的服务优化处理系统是很有必要的。

发明内容

[0003] 本发明的目的在于提供一种基于云平台的服务优化处理系统,以解决上述背景技术中提出的问题。

[0004] 为了解决上述技术问题,本发明提供如下技术方案:一种基于云平台的服务优化处理方法,包括:

- 进行云平台服务系统的架构设计管理;
- 对云平台服务系统进行消息传输及接口管理;
- 进一步对云平台服务系统进行分析控制;
- 进行云平台服务系统的采集反馈管理;

所述进行云平台服务系统的架构设计管理包括:

在OpenStack云平台服务系统中,使Sdocker通过调用OpenStack中Keystone、Glance以及Neutron组件暴露的API接口获取创建容器所需的镜像和网络资源,并通过调用Docker提供的API接口创建容器以及对容器进行更新处理。

[0005] 根据上述技术方案,所述对云平台服务系统进行消息传输及接口管理包括:

控制云平台服务系统组件间通过消息队列进行传输通信,包括:生产者(Producer)、交换器(Exchange)、绑定(Binding)、队列(Queue)和消费者(Consumer)。

[0006] 根据上述技术方案,所述对云平台服务系统进行消息传输及接口管理进一步包括:

通过Sdocker的sdock_API为用户提供RESTful API,使用户可以通过HTTP请求的方式方便快捷地查看容器及OpenStack中各个计算节点的相关信息。

[0007] 根据上述技术方案,所述进一步对云平台服务系统进行分析控制包括:

在云平台服务系统中,当用户开启Sdocker的负载均衡调度模式并指定容器资源

信息(CPU、内存、磁盘三种资源信息),向Sdocker请求创建容器,用户请求创建容器的命令里包含容器名称、容器镜像、容器网络信息以及容器的资源请求规格信息;

接收到容器调度请求信息后,sdoc_Scheduler组件首先对容器的资源信息进行解析,如果容器具有CPU、内存和磁盘信息,则对容器实施调度,在进行容器调度时均采用LBSM调度机制为容器选取最优计算节点和最优NUMA节点;

sdoc_Compute获取到的image信息后,根据容器的名称和容器资源请求信息,调用Docker提供的API接口创建并启动容器,Sdoc_Compute依托解析出的token,调用Neutron组件暴露的API接口创建port,并通过port在本地计算节点创建虚拟网卡,将该网卡挂载至该容器的网络命名空间,完成容器部署。

[0008] 根据上述技术方案,所述进行云平台服务系统的采集反馈管理包括:

对云平台服务系统的运行数据信息根据设定的周期进行采集,并将其传输至工作站处,使工作人员可以进行对应的监测分析;

用户可对云服务的使用进行意见反馈输入,为后续云平台服务系统维护优化提供有效的参考依据。

[0009] 根据上述技术方案,一种基于云平台的服务优化处理系统,包括:

设计传输模块,用于进行云平台服务系统的设计传输控制;
分析管理模块,用于进行云平台服务系统的分析优化管理;
采集反馈模块,用于进行云平台服务系统的信息采集反馈。

[0010] 根据上述技术方案,所述设计传输模块包括:

架构设计模块,用于对云平台服务系统进行架构设计管理;
消息传输模块,用于进行云平台服务系统组件间消息传输处理;
接口管理模块,用于进行云平台服务系统接口控制。

[0011] 根据上述技术方案,所述分析管理模块包括:

请求管理模块,用于进行云平台服务系统的负载均衡请求管理;
分析控制模块,用于进行云平台容器调度分析控制;
部署处理模块,用于进行云平台容器部署分析处理。

[0012] 根据上述技术方案,所述采集反馈模块包括:

信息采集模块,用于进行云平台服务系统的信息采集;
反馈输入模块,用于进行数据信息的反馈输入管理。

[0013] 与现有技术相比,本发明所达到的有益效果是:本发明,通过设置有设计传输模块、分析管理模块与采集反馈模块,既可以实现云平台服务系统各个组件之间消息的异步传输,提高容器调度效率,也可以降低云平台服务系统Sdocker体系各个组件之间的耦合度并提高sdocke体系的可用性,有效提升云平台服务系统的运行效率与准确性。

附图说明

[0014] 附图用来提供对本发明的进一步理解,并且构成说明书的一部分,与本发明的实施例一起用于解释本发明,并不构成对本发明的限制。在附图中:

图1是本发明实施例一提供的一种基于云平台的服务优化处理方法的流程图;

图2是本发明实施例二提供的一种基于云平台的服务优化处理系统的模块构成

图。

具体实施方式

[0015] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0016] 实施例一:图1为本发明实施例一提供的一种基于云平台的服务优化处理方法的流程图,本实施例可应用于云平台服务系统中,该方法可以由本发明实施例提供的一种基于云平台的服务优化处理系统来执行,该系统由多个软硬件模块组成,如图1所示,该方法具体包括以下步骤:

S101、进行云平台服务系统的架构设计管理;

示例性的,在本发明实施例中,在OpenStack云平台服务系统中,使Sdocker通过调用OpenStack中Keystone、Glance以及Neutron组件暴露的API接口获取创建容器所需的镜像和网络资源,并通过调用Docker提供的API接口创建容器以及对容器进行更新处理;该步骤中,Sdocker内部包含Sdoc_API、Sdoc_Scheduler、Sdoc_Compute、Sdoc_Recompute、Sdoc_Migration和Sdoc_Conductor六个组件,各个组件通过在本地节点提供配置文件实现与用户之间的交互,同时Sdocker的各个组件分工明确且耦合度低,可通过为Sdocker增加新组件的方式实现容器调度的可扩展性,响应更加复杂的容器调度场景,其中,Sdoc_API组件部署于OpenStack控制节点,主要负责用户请求参数处理、调度模式解析和令牌获取;Sdoc_Scheduler组件部署于控制节点,主要负责调度模式解析和容器调度;sdoc_Compute组件部署于OpenStack计算节点,不与Sdoc数据库直接交互,其主要负责资源获取与容器创建;Sdoc_Recompute组件部署于OpenStack计算节点,不与Sdoc数据库直接交互,其专为高资源利用率调度模式服务,主要负责容器监测与容器更新/迁移;Sdoc_Migration组件部署于OpenStack计算节点,不与Sdoc数据库交互,其专为高资源利用率调度模式服务,主要负责容器迁移与容器重新创建;sdoc_Conductor组件部署于控制节点,主要作为Sdoc_Compute组件和Sdoc_Recompute组件的数据库代理,避免位于OpenStack计算节点的sdoc_Compute组件以及Sdoc_Recompute组件远程访问位于控制节点的Sdoc数据库。

[0017] S102、对云平台服务系统进行消息传输及接口管理;

示例性的,在本发明实施例中,控制云平台服务系统组件间通过消息队列进行传输通信,包括:生产者(Producer)、交换器(Exchange)、绑定(Binding)、队列(Queue)和消费者(Consumer);该步骤中,Producer发送携带有routing key的消息至Exchange,Exchange负责将routing key同与其绑定的所有Queue的binding key进行对比,并将消息发送至匹配的Queue中保存,最后Consumer会从该Queue中获取消息,同时Exchange有Direct、Topic和Fanout三种常用类型,使用Direct类型的Exchange时,只有当队列的binding key与消息中的routing key完全相同时,才会将消息发送至该队列,使用Topic类型的Exchange时,binding key允许有通配符“#”(代表零个或多个单词)和”(代表一个单词),例如,routing key为“sdocker.jn.edu”的消息可以成功发送至binding key为“#.jn.edu”的队列或者binding key为“#.jn.*”的队列,但是不能发送至binding key为“*.edu”的队列,使用

Fanout类型的Exchange时,消息会被广播到与该Exchange绑定的所有队列,用户可根据需求,灵活选择合适的Exchange进行消息传输,通过该步骤中,既可以实现云平台服务系统各个组件之间消息的异步传输,提高容器调度效率,也可以降低云平台服务系统Sdocker体系各个组件之间的耦合度并提高sdocke体系的可用性。

[0018] 通过Sdocker的sdoc_API为用户提供RESTful API,使用户可以通过HTTP请求的方式方便快捷地查看容器及OpenStack中各个计算节点的相关信息;该步骤中,针对不同的调度模式,控制Sdoc_API组件对用户进行差异化显示,例如,当用户想要查看指定容器所有信息时,Sdoc_API组件会进行调度模式解析,只有在负载均衡调度模式下,且容器的资源需求种类为4时,才会向用户显示包含容器bandwidth字段在内的所有信息,否则,Sdoc_API组件向用户返回的容器信息不包含bandwidth字段,当用户想要查看指定计算节点所有信息时,Sdoc_API组件只有在负载均衡调度模式下,且容器的资源需求种类为4时,才会向用户显示包含计算节点“bandwidth_total”字段在内的所有信息,否则,Sdoc_API组件向用户返回的计算节点信息不包含“bandwidth_total”字段

S103、进一步对云平台服务系统进行分析控制;

示例性的,在本发明实施例中,在云平台服务系统中,当用户开启Sdocker的负载均衡调度模式并指定容器资源信息(CPU、内存、磁盘三种资源信息),向Sdocker请求创建容器,用户请求创建容器的命令里包含容器名称、容器镜像、容器网络信息以及容器的资源请求规格信息;该步骤中,Sdoc_API接收用户请求创建容器命令后,首先对用户的请求参数进行处理,校验用户的请求是否符合设定规则,如果用符合,则控制Sdoc_API在数据库中创建新的容器条目,否则,Sdoc_API会向用户返回参数出错信息,其中Sdoc_API通过调用Keystone组件暴露的API接口获取token,依托该token可以获得使用OpenStack中镜像和虚拟网络资源的权限,Sdoc_API将处理后的用户请求参数和获取到的token整合成容器调度请求信息,Sdoc_API将容器调度请求信息发送至消息队列,并返回给用户包含容器名称和容器镜像等在内的详细信息供用户查看,此时用户还不能查看容器的调度结果,Sdoc_Scheduler组件从消息队列中获取容器调度请求信息,为容器调度作准备。

[0019] 接收到容器调度请求信息后,sdoc_Scheduler组件首先对容器的资源信息进行解析,如果容器具有CPU、内存和磁盘信息,则对容器实施调度,在进行容器调度时均采用LBSM调度机制为容器选取最优计算节点和最优NUMA节点;该步骤中,Sdoc_Scheduler将处理后的用户请求参数、获取到的token以及容器调度结果整合成容器创建请求信息,Sdoc_Scheduler将容器创建请求信息发送至消息队列,最优计算节点的Sdoc_Compute组件从消息队列中获取容器创建请求信息,为容器创建作准备,Sdoc_Compute从容器创建请求信息中解析出token,依托该token,sdoc_Compute调用Glance组件暴露的API接口将容器所需image下载至本地计算节点。

[0020] sdoc_Compute获取到的image信息后,根据容器的名称和容器资源请求信息,调用Docker提供的API接口创建并启动容器,Sdoc_Compute依托解析出的token,调用Neutron组件暴露的API接口创建port,并通过port在本地计算节点创建虚拟网卡,将该网卡挂载至该容器的网络命名空间,完成容器部署;该步骤中,sdoc_Compute将创建好的容器相关信息整合为容器更新请求信息,并将容器更新请求信息发送至消息队列,sdoc_Conductor组件从消息队列中获取容器更新请求信息,依据容器更新请求消息中的容器调度结果等信息更新

Sdoc数据库中的容器条目和最优计算节点条目,此时容器部署流程完全结束,用户可通过Sdoc_API暴露的API接口查看完整的容器信息(此时的容器信息包含了容器调度结果),有效提升云平台服务系统的运行效率与准确性。

[0021] S104、进行云平台服务系统的采集反馈管理;

示例性的,在本发明实施例中,对云平台服务系统的运行数据信息根据设定的周期进行采集,并将其传输至工作台处,使工作人员可以进行对应的监测分析;

用户可对云服务的使用进行意见反馈输入,为后续云平台服务系统维护优化提供有效的参考依据。

[0022] 实施例二:本发明实施例二提供了一种基于云平台的服务优化处理系统,图2为本实施例二提供的一种基于云平台的服务优化处理系统的模块构成示意图,如图2所示,该系统包括:

设计传输模块,用于进行云平台服务系统的设计传输控制;

分析管理模块,用于进行云平台服务系统的分析优化管理;

采集反馈模块,用于进行云平台服务系统的信息采集反馈。

[0023] 在本发明的一些实施例中,设计传输模块包括:

架构设计模块,用于对云平台服务系统进行架构设计管理;

消息传输模块,用于进行云平台服务系统组件间消息传输处理;

接口管理模块,用于进行云平台服务系统接口控制。

[0024] 在本发明的一些实施例中,分析管理模块包括:

请求管理模块,用于进行云平台服务系统的负载均衡请求管理;

分析控制模块,用于进行云平台容器调度分析控制;

部署处理模块,用于进行云平台容器部署分析处理。

[0025] 在本发明的一些实施例中,采集反馈模块包括:

信息采集模块,用于进行云平台服务系统的信息采集;

反馈输入模块,用于进行数据信息的反馈输入管理。

[0026] 需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。

[0027] 最后应说明的是:以上所述仅为本发明的优选实施例而已,并不用于限制本发明,尽管参照前述实施例对本发明进行了详细的说明,对于本领域的技术人员来说,其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

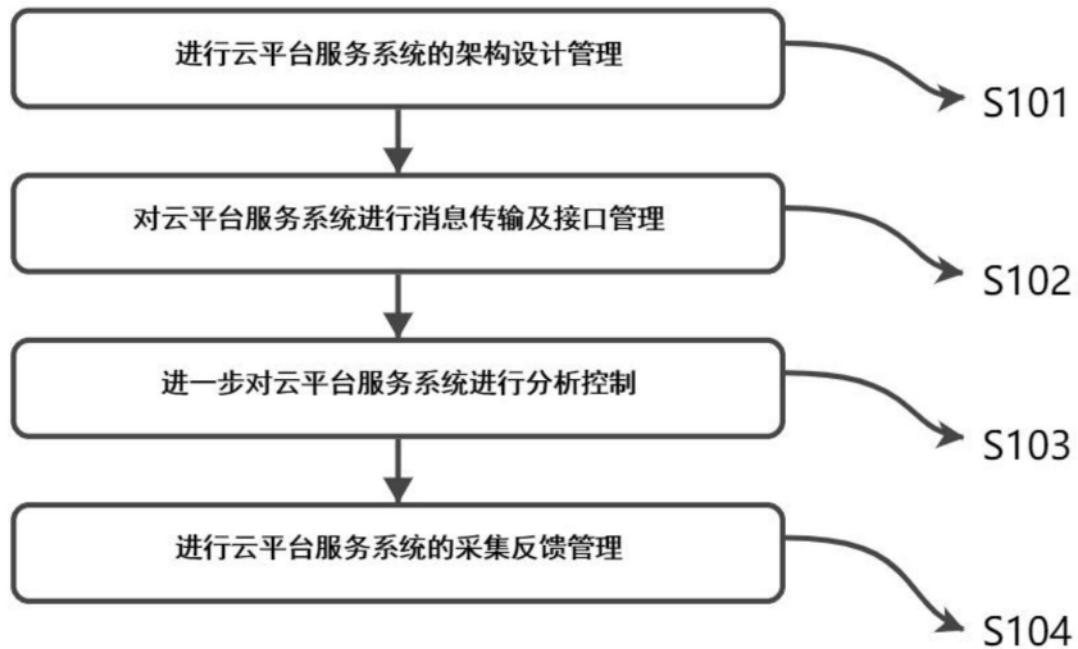


图1

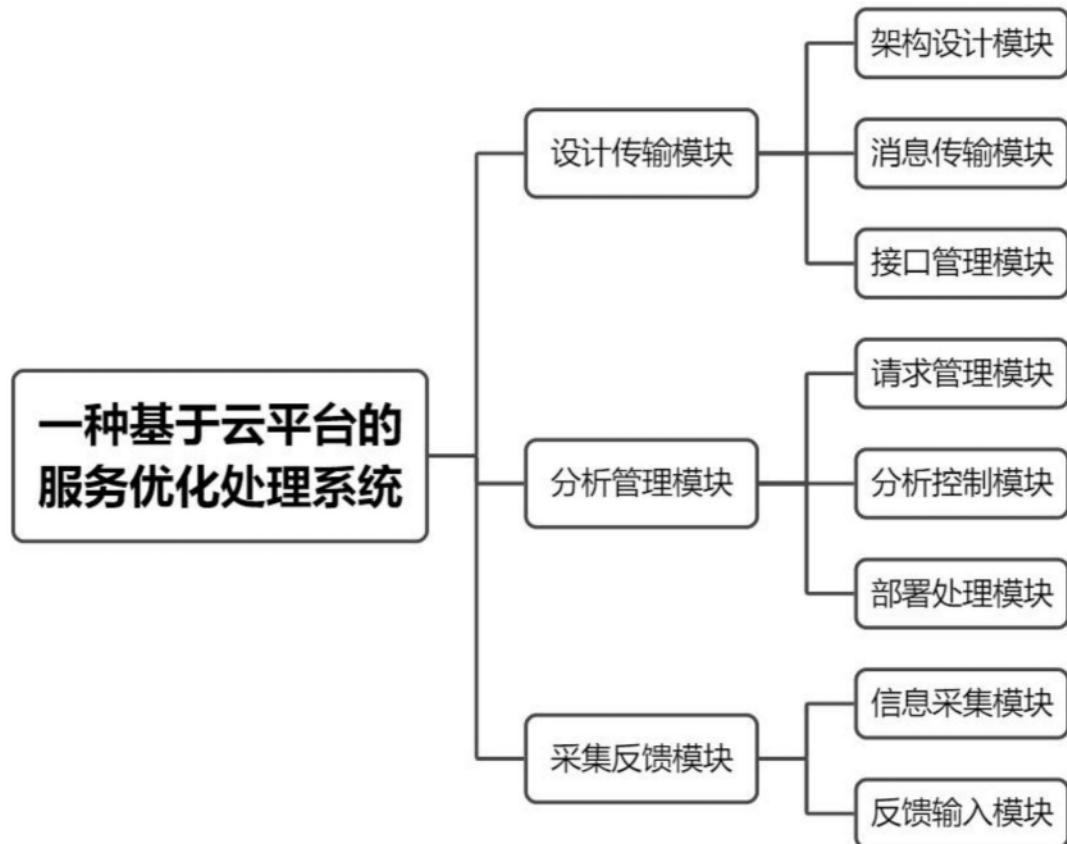


图2