



(12) 发明专利申请

(10) 申请公布号 CN 117675901 A

(43) 申请公布日 2024. 03. 08

(21) 申请号 202311618430.5

(22) 申请日 2023.11.30

(71) 申请人 浪潮软件科技有限公司

地址 250100 山东省济南市高新区浪潮路
1036号浪潮科技园

(72) 发明人 刘传相 李存冰 陈焕新

(74) 专利代理机构 济南信达专利事务所有限公
司 37100

专利代理师 阚恭勇

(51) Int. Cl.

H04L 67/51 (2022.01)

H04L 67/54 (2022.01)

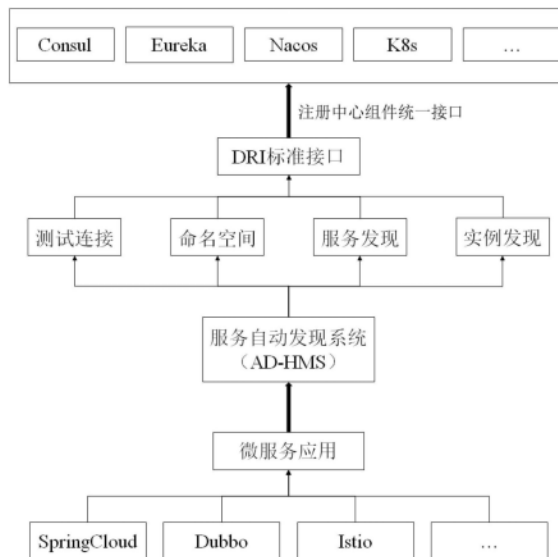
权利要求书1页 说明书3页 附图2页

(54) 发明名称

一种基于异构微服务自动发现的方法

(57) 摘要

本发明提供一种基于异构微服务自动发现的方法,属于微服务框架中的注册中心技术领域,本发明定义了一套分布式注册中心接口,构建了一个兼容多种微服务框架及多种注册中心的系统,支持SpringCloud、Dubbo、Isito等微服务框架,Consul、Eureka、Nacos、Kubernetes服务发现等服务注册中心。通过封装标准接口,实现自动获取不同类型微服务架构中的服务,消除了不同微服务框架的差异,打破了多种注册中心之间的联系壁垒,集成各注册中心工具的优势,为微服务自动发现提供了便捷性和统一性,以适应多样化的产品应用场景。



1. 一种基于异构微服务自动发现的方法,其特征在于,包括测试连接、命名空间、服务自动发现、实例自动发现,针对统一抽象注册中心接口,适配注册中心组件。
2. 根据权利要求1所述的方法,其特征在于,定义分布式注册中心接口,提供测试链接、命名空间、服务自动发现和实例自动发现四类接口,适配业界主流注册中心组件。
3. 根据权利要求2所述的方法,其特征在于,通过注册中心接口,实现自动服务发现,兼容异构微服务框架,以便于适配诸多微服务框架的需求。
4. 根据权利要求3所述的方法,其特征在于,通过实现DRI标准接口,实现接入各类型微服务框架,统一对外提供服务自动发现功能,屏蔽底层差异。
5. 根据权利要求4所述的方法,其特征在于,包括
步骤一:建立标准的分布式注册中心接口;
步骤二:AD-HMS实现封装的分布式注册中心接口。
6. 根据权利要求5所述的方法,其特征在于,所述步骤一,根据业界主流的注册中心组件,抽象出注册中心的统一功能方法;该方法包括以下四个方面:
 - 1) 测试连接:连接注册中心,获取注册中心组件健康状态;
 - 2) 命名空间:微服务在注册中心中通常存在于不同的命名空间,获取注册中心的命名空间,以便获取指定命名空间下的服务;
 - 3) 服务自动发现:从注册中心获取全部或指定命名空间下的服务;
 - 4) 实例自动发现:获取服务实例及权重,修改实例状态和权重。
7. 根据权利要求6所述的方法,其特征在于,根据统一功能方法,封装相应的接口DRI,DRI可以适配业界主流的注册中心组件,通过封装接口,可集成异构微服务框架,提供一致的API供用户使用。
8. 根据权利要求5或6所述的方法,其特征在于,所述步骤二,采用AD-HMS自动获取微服务的流程如下:
 - 1) 初始化配置,在应用程序的配置中添加自定义埋点代码,对接DRI;
 - 2) 请求服务,携带注册中心配置信息发起微服务应用创建,通过统一驱动工厂处理后发送请求,即可获取全部或指定命名空间下的服务信息,服务实例信息;
 - 3) 数据存储,存储服务信息,将服务信息本地持久化并定时更新,包括服务实例的IP,端口,状态,权重,接口;
 - 4) 信息展示与操作,基于不同的注册中心组件的数据与能力,提供个性化信息展示页面,差异化功能开放。

一种基于异构微服务自动发现的方法

技术领域

[0001] 本发明涉及微服务框架中的注册中心技术领域,尤其涉及一种基于异构微服务自动发现的方法。

背景技术

[0002] 微服务架构是一种面向服务的架构风格,通过将应用程序拆分为一组小型、松耦合的服务来提高系统的灵活性和可伸缩性。而注册中心是微服务架构中的关键组件之一,充当了服务发现和路由的关键角色,其作用是管理和维护各个微服务实例的元数据信息,包括服务名称、地址、版本等。它允许服务提供者将其服务注册到注册中心,同时也允许服务消费者查询并发现可用的服务实例,为服务提供者和服务消费者之间的通信提供支持。

[0003] 此外,因为不同微服务框架可能使用不同的注册中心组件,这导致了不同类型的服务实例难以统一概览和管理。传统的注册中心通常只适用于单一的微服务框架,对于多种微服务框架的服务实例集中管控存在诸多难题。因此,开发一种兼容、通用且高效的服务自动发现系统,以适应多种微服务框架的需求,具有重要的实际意义。

发明内容

[0004] 为了解决以上技术问题,本发明提供了一种基于异构微服务自动发现的方法,从多种不同注册中心自动获取服务实例,并且可持续监控服务实例的存在状况和健康状态。该方法提供了统一便捷的服务实例发现和监控技术,通过该技术将自动发现的服务实例集中,从而使开发人员无需关注微服务使用的注册中心,能够方便开发人员发现服务并监视运行情况,降低服务发现的复杂度,提高微服务管理的效率。

[0005] 本发明的技术方案是:

[0006] 一种基于异构微服务自动发现的方法,包括

[0007] 测试连接、命名空间、服务自动发现、实例自动发现,针对统一抽象注册中心接口,适配业界主流注册中心组件。

[0008] 进一步的,

[0009] 抽象定义一套标准的分布式注册中心接口,主要提供测试链接、命名空间、服务自动发现和实例自动发现四类接口,适配业界主流注册中心组件。

[0010] 通过一套通用且高效的注册中心接口,实现自动服务发现,兼容异构微服务框架,以便于适配诸多微服务框架的需求。

[0011] 通过实现DRI标准接口,实现接入各类型微服务框架,统一对外提供服务自动发现功能,屏蔽底层差异。

[0012] 进一步的,

[0013] 包括

[0014] 步骤一:建立标准的分布式注册中心接口;

[0015] 根据业界主流的注册中心组件,抽象出注册中心的统一功能方法;该方法包括以

下四个方面：

[0016] 1) 测试连接:连接注册中心,获取注册中心组件健康状态;

[0017] 2) 命名空间:微服务在注册中心中通常存在于不同的命名空间,获取注册中心的命名空间,以便获取指定命名空间下的服务;

[0018] 3) 服务自动发现:从注册中心获取全部或指定命名空间下的服务;

[0019] 4) 实例自动发现:获取服务实例及权重,修改实例状态和权重。

[0020] 根据统一功能方法,封装相应的接口DRI,DRI可以适配业界主流的注册中心组件,通过封装接口,可集成异构微服务框架,提供一致的API供用户使用。

[0021] 步骤二:AD-HMS实现封装的分布式注册中心接口。

[0022] 采用AD-HMS自动获取微服务的流程如下:

[0023] 1) 初始化配置,在应用程序的配置中添加自定义埋点代码,对接DRI;

[0024] 2) 请求服务,携带注册中心配置信息发起微服务应用创建,通过统一驱动工厂处理后发送请求,即可获取全部或指定命名空间下的服务信息,服务实例信息;

[0025] 3) 数据存储,存储服务信息,将服务信息本地持久化并定时更新,包括服务实例的IP,端口,状态,权重,接口;

[0026] 4) 信息展示与操作,基于不同的注册中心组件的数据与能力,提供个性化信息展示页面,差异化功能开放。

[0027] 本发明的有益效果是

[0028] 本发明兼容Consul、Eureka、Nacos、Kubernetes服务发现等主流分布式系统架构的注册中心,可实现SpringCloud、Dubbo、Isito等异构微服务类型服务自动发现,通过某种技术自动地实现对异构微服务框架中的服务实例发现,以提供统一、便捷、可靠的服务实例管理能力,支持多种微服务类型应用的服务自动发现。

附图说明

[0029] 图1是本发明的结构工作流程框图;

[0030] 图2是本发明的整体技术架构示意图。

具体实施方式

[0031] 为使本发明实施例的目的、技术方案和优点更加清楚,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的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。

[0032] 本发明一种实现了异构微服务框架的服务自动发现方法,定义了一套分布式注册中心接口(DistributedRegistry Interface,简称DRI),构建了一个兼容多种微服务框架及多种注册中心的系统,支持SpringCloud、Dubbo、Isito等微服务框架,Consul、Eureka、Nacos、Kubernetes服务发现等服务注册中心。通过封装标准接口,实现自动获取不同类型微服务架构中的服务,消除了不同微服务框架的差异,打破了多种注册中心之间的联系壁垒,集成各注册中心工具的优势,为微服务自动发现提供了便捷性和统一性,以适应多样化的产品应用场景。

[0033] 本发明首先根据主流的注册中心组件,抽象出注册中心的统一功能方法。包括测试连接、命名空间、服务自动发现、实例自动发现。然后针对统一抽象注册中心接口,适配业界主流注册中心组件。

[0034] 步骤一:建立标准的分布式注册中心接口(DRI)

[0035] 首先,根据业界主流的注册中心组件,抽象出注册中心的统一功能方法,该方法包括以下四个方面:

[0036] 1) 测试连接:连接注册中心,获取注册中心组件健康状态。

[0037] 2) 命名空间:微服务在注册中心中通常存在于不同的命名空间,获取注册中心的命名空间,以便获取指定命名空间下的服务。

[0038] 3) 服务自动发现:从注册中心获取全部或指定命名空间下的服务。

[0039] 4) 实例自动发现:获取服务实例及权重,修改实例状态和权重。

[0040] 接下来,根据上述抽象出的统一功能方法,封装相应的接口DRI,DRI可以适配业界主流的注册中心组件,包括但不限于Consul、Eureka、Nacos、Kubernetes服务发现等,通过封装接口,可集成异构微服务框架,提供一致的API供用户使用。

[0041] 步骤二:AD-HMS实现封装的分布式注册中心接口(DRI)

[0042] 采用AD-HMS自动获取微服务的流程如下:

[0043] 1) 初始化配置。在应用程序的配置中添加自定义埋点代码,对接DRI;

[0044] 2) 请求服务。携带注册中心配置信息发起微服务应用创建,通过统一驱动工厂处理后发送请求,即可获取全部或指定命名空间下的服务信息,服务实例信息。

[0045] 3) 数据存储。存储服务信息,将服务信息本地持久化并定时更新,包括但不限于服务实例的IP,端口,状态,权重,接口。

[0046] 4) 信息展示与操作。基于不同的注册中心组件的数据与能力,提供个性化信息展示页面,差异化功能开放。

[0047] 采用此技术,对接的注册中心组件包括:Consul、Eureka、Nacos等,以及Kubernetes服务发现,实现了异构微服务框架的服务自动发现功能。

[0048] 以上所述仅为本发明的较佳实施例,仅用于说明本发明的技术方案,并非用于限定本发明的保护范围。凡在本发明的精神和原则之内所做的任何修改、等同替换、改进等,均包含在本发明的保护范围内。

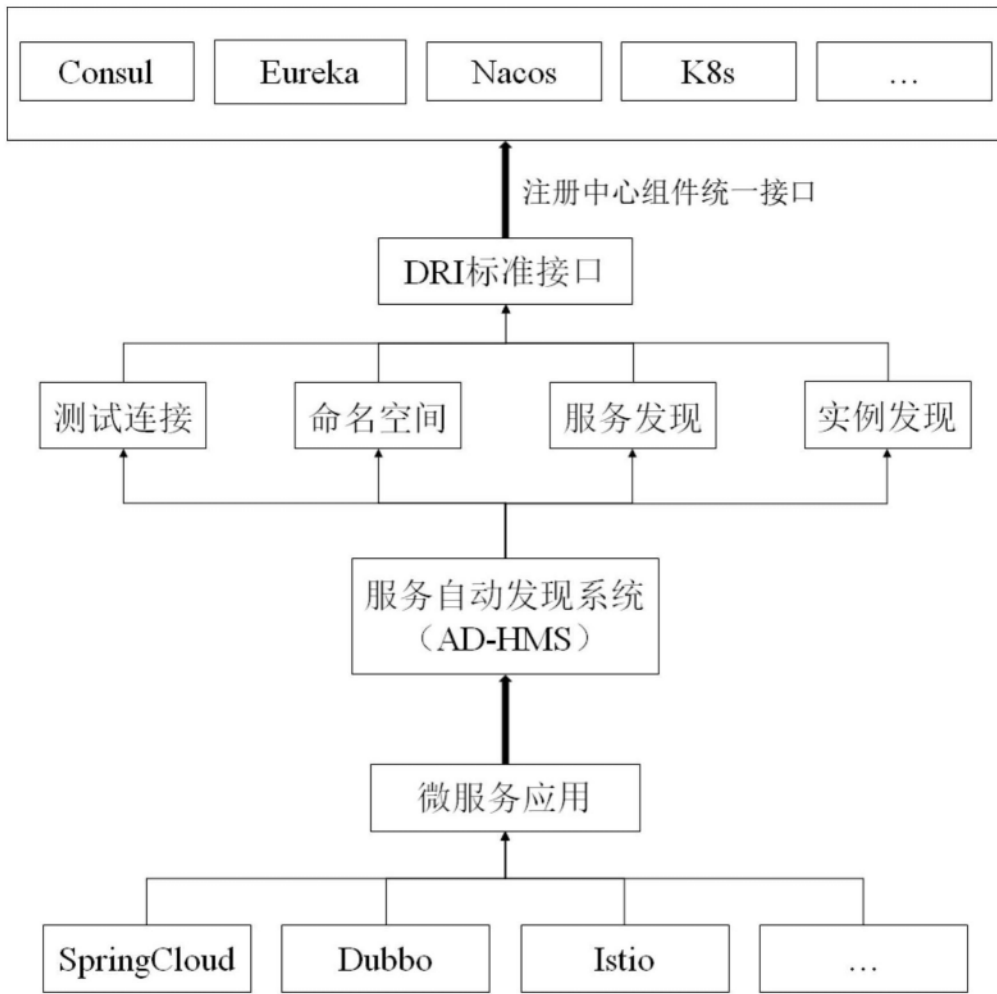


图1

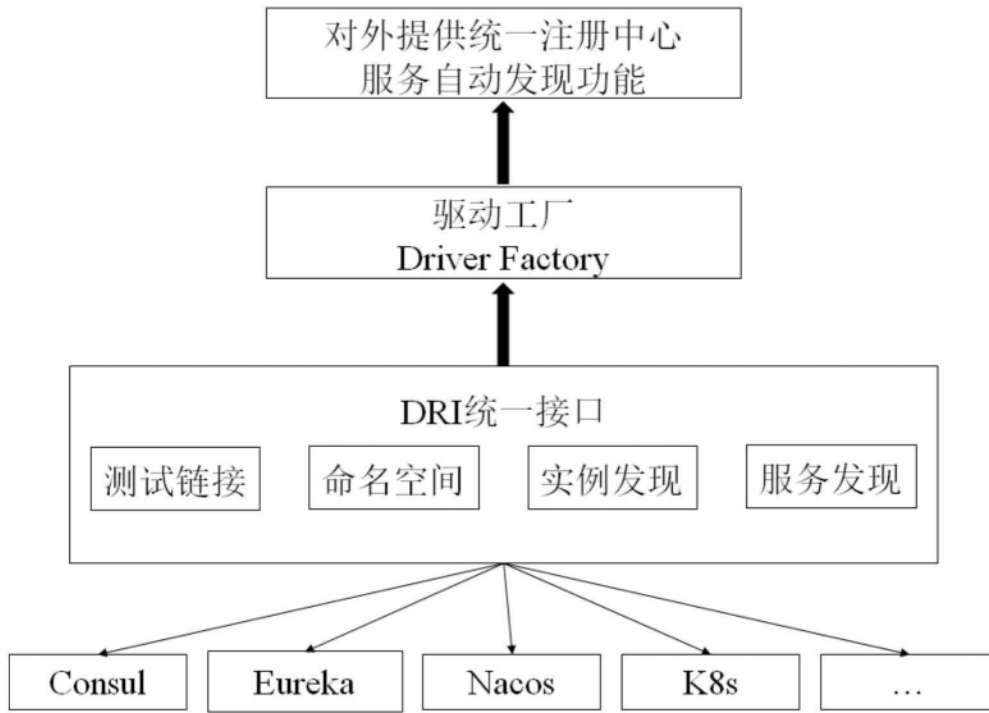


图2