(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112328362 A
(43)申请公布日 2021.02.05

(21)申请号 202011221265.6

(22)申请日 2020.11.03

(71)申请人 浪潮云信息技术股份公司
地址 250100 山东省济南市高新区浪潮路1036号浪潮科技园S01号楼

(72)发明人 于春钰 张晖 高传集 孙兴艳 王刚

(74)专利代理机构 济南信达专利事务所有限公司 37100
代理人 冯春连

(51)Int.Cl.
G06F 9/455(2006.01)
G06F 9/445(2018.01)

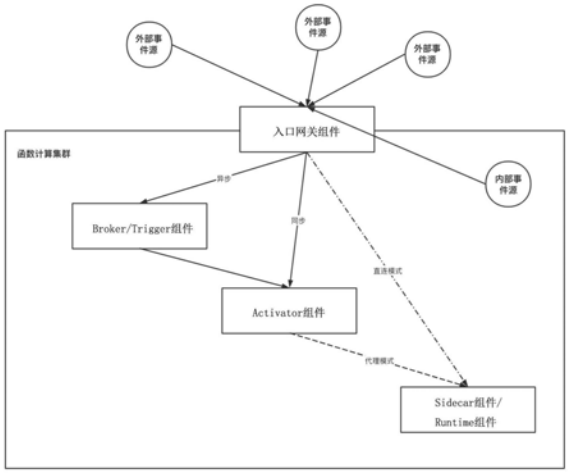
权利要求书2页 说明书5页 附图2页

(54)发明名称

一种基于容器技术实现函数计算服务的方法

(57)摘要

本发明公开一种基于容器技术实现函数计算服务的方法，涉及无服务器计算技术领域，该方法基于入口网关组件、Broker/Trigger组件、Activator组件、Sidecar组件，其实现内容包括：使用容器技术构建多租户场景下的函数计算服务，使用入口网关组件接收外部执行函数的事件请求，并按照特定的规则将请求转发，使用Broker/Trigger组件接收并转发函数事件，达到异步处理的功能，使用Activator组件控制请求模式，在函数实例未完全启动时选择代理模式处理事件请求，在函数可以正常提供服务时选择直连模式将请求直接导向函数实例，使用Sidecar组件对单个函数实例进行监控与管理，最终实现事件驱动、按需付费、高可用、自动扩展的多租户函数计算服务。本发明可以优化函数的冷启动，做到快速扩缩容和响应。



CN 112328362 A

1.一种基于容器技术实现函数计算服务的方法,其特征在于,基于入口网关组件、Broker/Trigger组件、Activator组件、Sidecar组件,该方法:

使用容器技术构建多租户场景下的函数计算服务,

使用入口网关组件接收外部执行函数的事件请求,并按照特定的规则将请求转发,

使用Broker/Trigger组件接收并转发函数事件,达到异步处理的功能,

使用Activator组件控制请求模式,在函数实例未完全启动时选择代理模式处理事件请求,在函数可以正常提供服务时选择直连模式将请求直接导向函数实例,

使用Sidecar组件对单个函数实例进行监控与管理,

最终实现事件驱动、按需付费、高可用、自动扩展的多租户函数计算服务。

2.根据权利要求1所述的一种基于容器技术实现函数计算服务的方法,其特征在于,函数计算服务的调用流程为:

(1)由函数计算集群的内部事件源或外部事件源产生事件,产生的事件即为函数调用请求;

(2)事件到达入口网关组件,入口网关组件对事件进行权限校验;

(3)校验通过后,事件直接或间接进入函数计算集群;

(4)事件在函数计算集群内传递,Activator组件根据函数实例是否能正常提供服务控制事件的请求模式,并转发给函数实例,该函数实例由Sidecar组件进行监控与管理。

3.根据权利要求2所述的一种基于容器技术实现函数计算服务的方法,其特征在于,步骤(1)产生的事件分为同步事件和异步事件,

执行步骤(3)时,同步事件通过校验后,直接进入函数计算集群,

异步事件通过校验后,发送至Broker/Trigger组件,进行进一步的分发进入函数计算集群。

4.根据权利要求3所述的一种基于容器技术实现函数计算服务的方法,其特征在于,执行步骤(4)时,事件在函数计算集群内的传递分为直连模式和代理模式,

Activator组件根据函数实例是否能正常提供服务控制事件的请求模式,事件在函数计算集群内选择直连模式时,函数计算集群将事件直接转发给函数实例,事件在函数计算集群内选择代理模式时,事件会在Activator组件上进行中转,等待函数实例可以正常处理请求时,再将事件转发给函数实例。

5.根据权利要求4所述的一种基于容器技术实现函数计算服务的方法,其特征在于,所述Broker/Trigger组件包含了一个用于暂存事件请求的消息队列,该消息队列可以使用In Memory Channel内存管道来实现,也可以使用开源方案Kafka的消息队列来实现;

异步事件通过校验后,事件通过入口网关组件进入到Broker/Trigger组件的消息队列中,随后函数计算集群内部的事件源就会收到事件请求成功的返回结果。

6.根据权利要求5所述的一种基于容器技术实现函数计算服务的方法,其特征在于,所述Broker/Trigger组件接收函数事件,并为每一个租户提供一个特定的URL用来发送事件,如果函数需要消费对应的事件,则需要创建一个Trigger,指定监听某个Broker接收到的事件,并可以通过事件中的Attributes来过滤出自己需要的事件。

7.根据权利要求6所述的一种基于容器技术实现函数计算服务的方法,其特征在于,事件进入函数计算集群后,由Istio提供函数计算集群内流量的管理,流量到达函数之前,函

数实例可能还不能正常提供服务，这时Activator组件就会将请求暂时挂起，等待函数实例正常启动后，再将流量导入到函数实例；

在直连模式和代理模式切换的过程中，Activator组件通过修改函数实例的Kubernetes Service对应的Endpoint来实现，其中，直连模式将Endpoint设置为函数实例的IP，代理模式将Endpoint设置为Activator组件的IP。

8.根据权利要求7所述的一种基于容器技术实现函数计算服务的方法，其特征在于，事件到达函数实例后，经过Sidecar组件，Sidecar组件负责对函数实例进行监控与管理，Sidecar组件还负责对函数代码的拉取和更新操作，以及函数超时的控制，最终由用户函数执行并返回结果。

9.根据权利要求8所述的一种基于容器技术实现函数计算服务的方法，其特征在于，事件到达函数实例后，还由Sidecar组件转发至Runtime组件，Runtime组件负责加载函数的运行时，并封装函数的入参，以及处理函数的返回值。

10.根据权利要求1或2所述的一种基于容器技术实现函数计算服务的方法，其特征在于，所述入口网关组件基于开源方案Kong已有的网关功能，增加相应的插件实现权限校验功能和请求规则转发功能，其中，

权限校验功能基于HTTP请求的Headers进行，拿到Token与后端接口进行校验，验证通过后方可将流量转入集群内，

请求规则转发功能基于HTTP请求的Path进行。

一种基于容器技术实现函数计算服务的方法

技术领域

[0001] 本发明涉及无服务器计算技术领域，具体的说是一种基于容器技术实现函数计算服务的方法。

背景技术

[0002] 云计算经历了从IDC->IaaS->PaaS->Serverless/FaaS的发展历程。过去十多年的云计算的历程，其实是一个“去基础架构”的过程。这个过程让用户可以更快速、更简单、更高效地将想法变成应用，变成在线的服务。Serverless架构即“无服务器”架构，它是一种全新的架构方式，是云计算时代一种革命性的架构模式。Serverless让用户可以将关注点放到具体的业务功能上，而不是底层的计算资源上。

[0003] 在传统的场景里，当用户完成了应用开发后，软件应用将被部署到指定的运行环境，用户会申请一定数量、一定规格(包含一定数量的CPU、内存及存储空间)的服务器以满足该应用的正常运行。当应用上线后，根据实际的运营情况，用户可能会申请更多的服务器资源进行扩容，以应对更高的访问量。在Serverless架构下，情况则截然不同。当用户完成应用开发后，软件应用将被部署到指定的运行环境，这个运行环境不再是具体的一台或多台服务器，而是支持Serverless的云计算平台。有客户端请求到达或特定事件发生时，云计算平台负责将应用部署到某台Serverless云计算平台的主机中。Serverless云计算平台保证该主机提供应用正常运行所需的计算资源。在访问量升高时，云计算平台动态地增加应用的部署实例。当应用空闲一段时间后，云计算平台自动将应用从主机中卸载，并回收资源。

[0004] 首先要明确的一点是，Serverless是一种软件的架构理念。它的核心思想是让作为计算资源的服务器不再成为用户所关注的一种资源。其目的是提高应用交付的效率，降低应用运营的工作量和成本。但是，要实现Serverless架构的落地，需要一些实实在在的工具和框架作为有力的技术支撑和基础。现有的框架的功能都不够全面，很多框架只是实现了事件驱动、自动扩展、请求分发等部分功能，无法实际用于生产环境；而且，函数代码动态加载、函数冷启动、函数间的安全隔离等问题，也是现在未解决的难题。

发明内容

[0005] 本发明针对目前技术发展的需求和不足之处，提供一种基于容器技术实现函数计算服务的方法。

[0006] 本发明的一种基于容器技术实现函数计算服务的方法，解决上述技术问题采用的技术方案如下：

[0007] 一种基于容器技术实现函数计算服务的方法，基于入口网关组件、Broker/Trigger组件、Activator组件、Sidecar组件，该方法：

[0008] 使用容器技术构建多租户场景下的函数计算服务，

[0009] 使用入口网关组件接收外部执行函数的事件请求，并按照特定的规则将请求转

发，

[0010] 使用Broker/Trigger组件接收并转发函数事件，达到异步处理的功能，

[0011] 使用Activator组件控制请求模式，在函数实例未完全启动时选择代理模式处理事件请求，在函数可以正常提供服务时选择直连模式将请求直接导向函数实例，

[0012] 使用Sidecar组件对单个函数实例进行监控与管理，

[0013] 最终实现事件驱动、按需付费、高可用、自动扩展的多租户函数计算服务。

[0014] 可选的，所涉及函数计算服务的调用流程为：

[0015] (1)由函数计算集群的内部事件源或外部事件源产生事件，产生的事件即为函数调用请求；

[0016] (2)事件到达入口网关组件，入口网关组件对事件进行权限校验；

[0017] (3)校验通过后，事件直接或间接进入函数计算集群；

[0018] (4)事件在函数计算集群内传递，Activator组件根据函数实例是否能正常提供服务控制事件的请求模式，并转发给函数实例，该函数实例由Sidecar组件进行监控与管理。

[0019] 进一步可选的，所涉及步骤(1)产生的事件分为同步事件和异步事件，

[0020] 执行步骤(3)时，同步事件通过校验后，直接进入函数计算集群，

[0021] 异步事件通过校验后，发送至Broker/Trigger组件，进行进一步的分发进入函数计算集群。

[0022] 进一步可选的，执行步骤(4)时，事件在函数计算集群内的传递分为直连模式和代理模式，

[0023] Activator组件根据函数实例是否能正常提供服务控制事件的请求模式，事件在函数计算集群内选择直连模式时，函数计算集群将事件直接转发给函数实例，事件在函数计算集群内选择代理模式时，事件会在Activator组件上进行中转，等待函数实例可以正常处理请求时，再将事件转发给函数实例。

[0024] 进一步可选的，所涉及Broker/Trigger组件包含了一个用于暂存事件请求的消息队列，该消息队列可以使用In Memory Channel内存管道来实现，也可以使用开源方案Kafka的消息队列来实现；

[0025] 异步事件通过校验后，事件通过入口网关组件进入到Broker/Trigger组件的消息队列中，随后函数计算集群内部的事件源就会收到事件请求成功的返回结果。

[0026] 进一步可选的，所涉及Broker/Trigger组件接收函数事件，并为每一个租户提供一个特定的URL用来发送事件，如果函数需要消费对应的事件，则需要创建一个Trigger，指定监听某个Broker接收到的事件，并可以通过事件中的Attributes来过滤出自己需要的事件。

[0027] 进一步可选的，所涉及事件进入函数计算集群后，由Istio提供函数计算集群内流量的管理，流量到达函数之前，函数实例可能还不能正常提供服务，这时Activator组件就会将请求暂时挂起，等待函数实例正常启动后，再将流量导入到函数实例；

[0028] 在直连模式和代理模式切换的过程中，Activator组件通过修改函数实例的Kubernetes Service对应的Endpoint来实现，其中，直连模式将Endpoint设置为函数实例的IP，代理模式将Endpoint设置为Activator组件的IP。

[0029] 可选的，所涉及事件到达函数实例后，经过Sidecar组件，Sidecar组件负责对函数

实例进行监控与管理,Sidecar组件还负责对函数代码的拉取和更新操作,以及函数超时的控制,最终由用户函数执行并返回结果。

[0030] 进一步可选的,所涉及事件到达函数实例后,还由Sidecar组件转发至Runtime组件,Runtime组件负责加载函数的运行时,并封装函数的入参,以及处理函数的返回值。

[0031] 可选的,所涉及入口网关组件基于开源方案Kong已有的网关功能,增加相应的插件实现权限校验功能和请求规则转发功能,其中,

[0032] 权限校验功能基于HTTP请求的Headers进行,拿到Token与后端接口进行校验,验证通过后方可将流量转入集群内,

[0033] 请求规则转发功能基于HTTP请求的Path进行。

[0034] 本发明的一种基于容器技术实现函数计算服务的方法,与现有技术相比具有的有益效果是:

[0035] 本发明基于容器技术,结合入口网关组件、Broker/Trigger组件、Activator组件、Sidecar组件,可以实现函数流量的统一管理、分发,实现函数访问的统一鉴权、监控,实现函数间的安全隔离,实现函数的快速响应,最终实现事件驱动、按需付费、高可用、自动扩展的多租户函数计算服务。

附图说明

[0036] 附图1是本发明的方法流程示意图;

[0037] 附图2是本发明中事件到达函数实例内部后的处理流程图。

具体实施方式

[0038] 为使本发明的技术方案、解决的技术问题和技术效果更加清楚明白,以下结合具体实施例,对本发明的技术方案进行清楚、完整的描述。

[0039] 实施例一:

[0040] 本实施例提出一种基于容器技术实现函数计算服务的方法,该方法基于入口网关组件、Broker/Trigger组件、Activator组件、Sidecar组件,具体实现内容包括:

[0041] 使用容器技术构建多租户场景下的函数计算服务,

[0042] 使用入口网关组件接收外部执行函数的事件请求,并按照特定的规则将请求转发,

[0043] 使用Broker/Trigger组件接收并转发函数事件,达到异步处理的功能,

[0044] 使用Activator组件控制请求模式,在函数实例未完全启动时选择代理模式处理事件请求,在函数可以正常提供服务时选择直连模式将请求直接导向函数实例,

[0045] 使用Sidecar组件对单个函数实例进行监控与管理,

[0046] 最终实现事件驱动、按需付费、高可用、自动扩展的多租户函数计算服务。

[0047] 针对上述实现内容,结合附图1,本实施例所涉及函数计算服务的调用流程为:

[0048] (1)由函数计算集群的内部事件源或外部事件源产生事件,产生的事件即为函数调用请求。

[0049] (2)事件到达入口网关组件,入口网关组件对事件进行权限校验。本实施例中,所涉及入口网关组件基于开源方案Kong已有的网关功能,增加相应的插件实现权限校验功能

和请求规则转发功能，其中：权限校验功能基于HTTP请求的Headers进行，拿到Token与后端接口进行校验，验证通过后方可将流量转入集群内，请求规则转发功能基于HTTP请求的Path进行。

[0050] (3)事件可以是同步事件/异步事件，

[0051] (3a)若同步事件通过校验后，直接进入函数计算集群，执行步骤(4)；

[0052] (3b)若异步事件通过校验后，发送至Broker/Trigger组件，进行进一步的分发进入函数计算集群，执行步骤(4)。

[0053] 本实施例的Broker/Trigger组件包含了一个用于暂存事件请求的消息队列，该消息队列可以使用In Memory Channel内存管道来实现，也可以使用开源方案Kafka的消息队列来实现。此时，异步事件通过校验后，事件通过入口网关组件进入到Broker/Trigger组件的消息队列中，随后函数计算集群内部的事件源就会收到事件请求成功的返回结果。

[0054] 本实施例执行上述步骤时，所涉及Broker/Trigger组件接收函数事件，并为每一个租户提供一个特定的URL用来发送事件，如果函数需要消费对应的事件，则需要创建一个Trigger，指定监听某个Broker接收到的事件，并可以通过事件中的Attributes来过滤出自己需要的事件。

[0055] (4)事件在函数计算集群内传递，Activator组件根据函数实例是否能正常提供服务控制事件的请求模式：直连模式或代理模式，

[0056] (4a)若事件在函数计算集群内选择直连模式时，函数计算集群将事件直接转发给函数实例，该函数实例由Sidecar组件进行监控与管理；

[0057] (4b)若事件在函数计算集群内选择代理模式时，事件会在Activator组件上进行中转，等待函数实例可以正常处理请求时，再将事件转发给函数实例，该函数实例由Sidecar组件进行监控与管理。

[0058] 本实施例执行上述步骤时，所涉及事件进入函数计算集群后，由Istio提供函数计算集群内流量的管理，流量到达函数之前，函数实例可能还不能正常提供服务，这时Activator组件就会将请求暂时挂起，等待函数实例正常启动后，再将流量导入到函数实例。

[0059] 在直连模式和代理模式切换的过程中，Activator组件通过修改函数实例的Kubernetes Service对应的Endpoint来实现，其中，直连模式将Endpoint设置为函数实例的IP，代理模式将Endpoint设置为Activator组件的IP。

[0060] 本实施例执行上述步骤时，结合附图2，所涉及事件到达函数实例后，经过Sidecar组件，Sidecar组件负责对函数实例进行监控与管理，Sidecar组件还负责对函数代码的拉取和更新操作，以及函数超时的控制，最终由用户函数执行并返回结果。

[0061] 结合附图2，所涉及事件到达函数实例后，还由Sidecar组件转发至Runtime组件，Runtime组件负责加载函数的运行时，并封装函数的入参，以及处理函数的返回值。

[0062] 综上可知，采用本发明的一种基于容器技术实现函数计算服务的方法，可以实现函数流量的统一管理、分发，实现函数访问的统一鉴权、监控，实现函数间的安全隔离，实现函数的快速响应。

[0063] 以上应用具体个例对本发明的原理及实施方式进行了详细阐述，这些实施例只是用于帮助理解本发明的核心技术内容。基于本发明的上述具体实施例，本技术领域的技术

人员在不脱离本发明原理的前提下,对本发明所作出的任何改进和修饰,皆应落入本发明的专利保护范围。

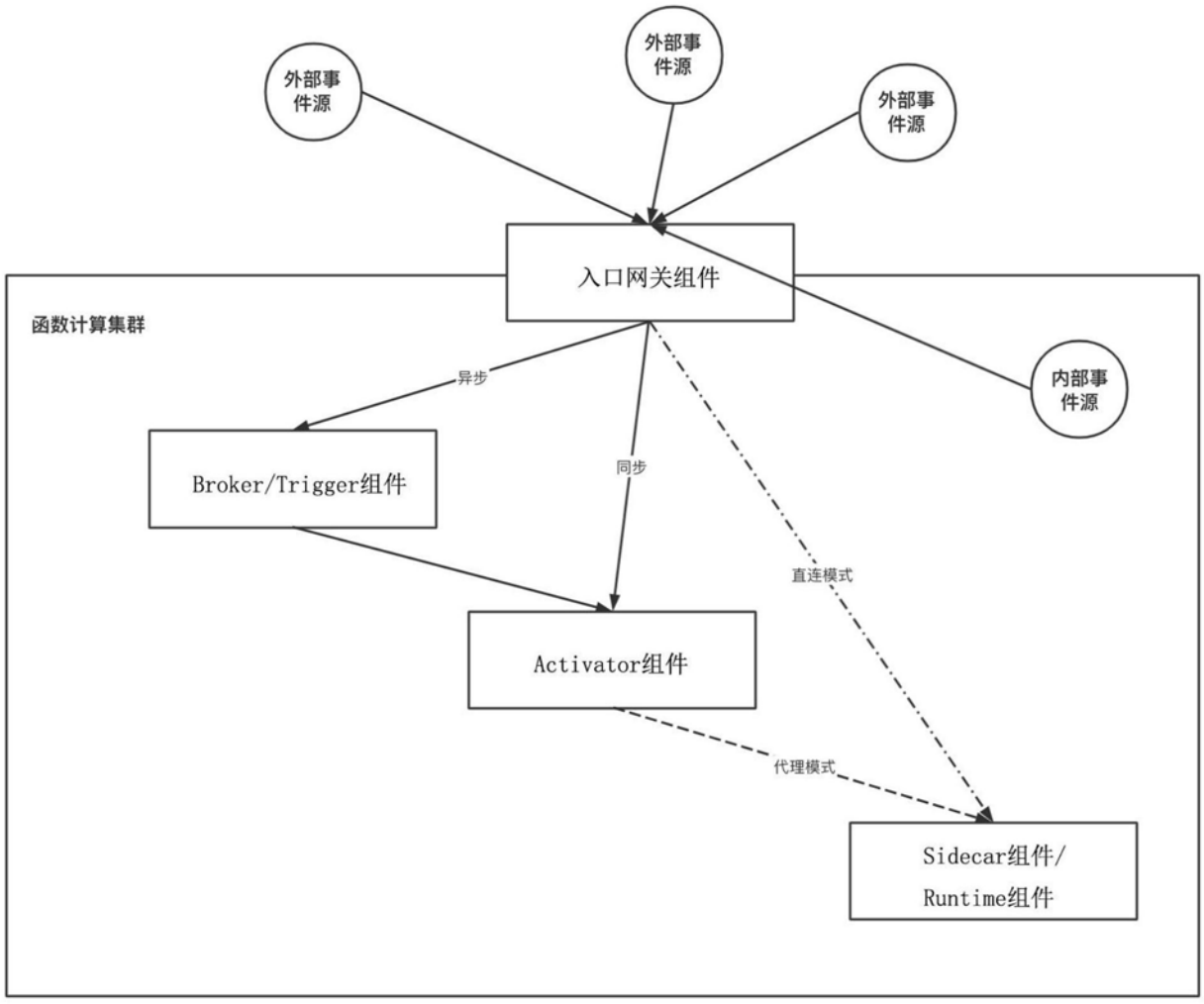

外部事件源
外部事件源
外部事件源
入口网关组件
函数计算集群
异步
同步
直连模式
内部事件源
Broker/Trigger组件
Activator组件
代理模式
Sidecar组件/
Runtime组件

图1

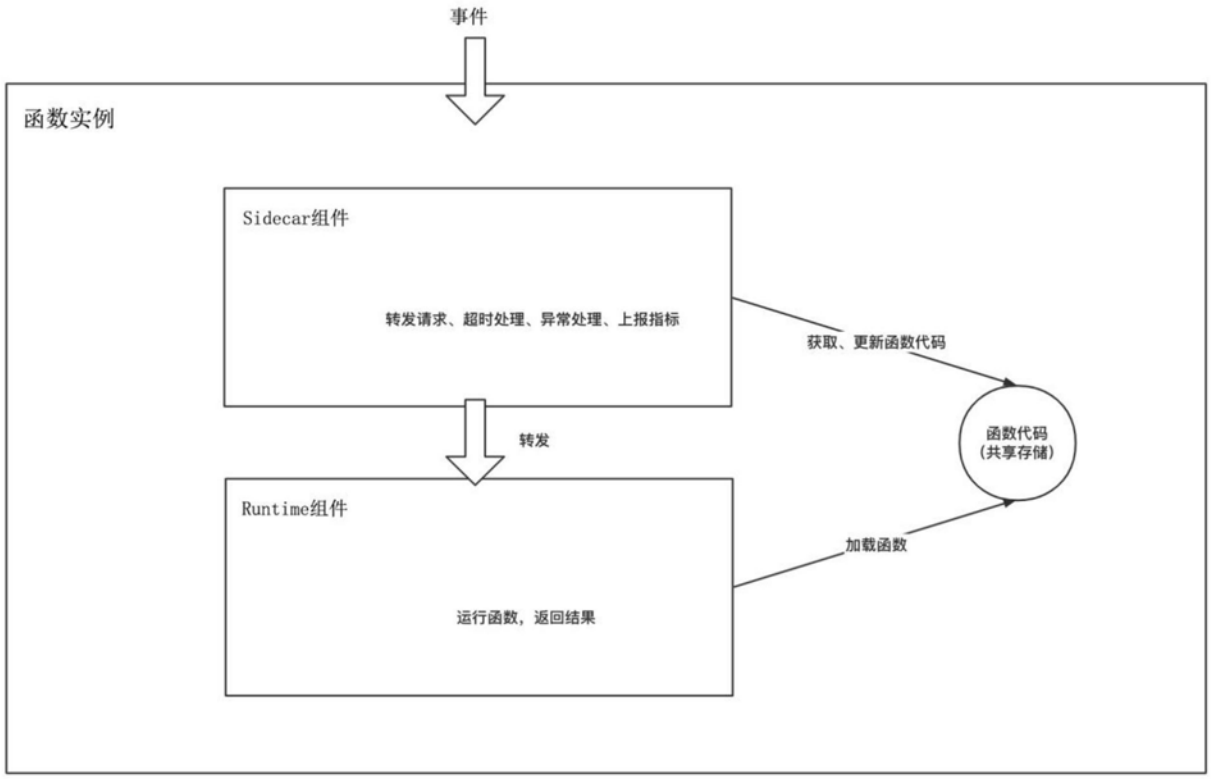

事件
函数实例
Sidecar组件
转发请求、超时处理、异常处理、上报指标
获取、更新函数代码
函数代码
(共享存储)
转发
Runtime组件
加载函数
运行函数，返回结果

图2