(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114444136 B
(45)授权公告日 2024.08.30

(21)申请号 202210119038.5

(22)申请日 2022.02.08

(65)同一申请的已公布的文献号
申请公布号 CN 114444136 A

(43)申请公布日 2022.05.06

(73)专利权人 上海富数科技有限公司
地址 201802 上海市嘉定区银翔路655号1幢4层416室

(72)发明人 朱崇炳 卞阳 赵东 尤志强

(74)专利代理机构 北京慧加伦知识产权代理有限公司 16035
专利代理师 兰海叶

(51)Int.Cl.
G06F 21/71(2013.01)
G06F 9/54(2006.01)

(56)对比文件
CN 101232626 A,2008.07.30
CN 112200505 A,2021.01.08
AI浩.“利用函数指针实现C的回调函数,实现调用者和底层驱动的解耦 第二种方式”.《CSDN》.2020,

审查员 江梓琴

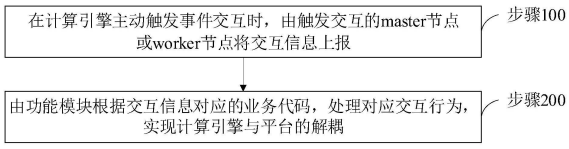
权利要求书2页 说明书9页 附图3页

(54)发明名称

一种计算引擎与平台解耦的方法及系统

(57)摘要

本申请提供一种计算引擎与平台解耦的方法及系统,采用driver-master-worker的结构来设计计算引擎,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互信息上报,使得功能模块能够根据交互信息对应的业务代码,处理对应交互行为,即可实现计算引擎主动触发交互的解耦,解耦之后,计算引擎被集成时,由平台厂家进行适配,计算引擎不需要做适配。因此,本申请实施例的方法在互联互通的使用场景中,计算引擎可以通过通用的集成方法被集成。



CN 114444136 B

1.一种计算引擎与平台解耦的方法,其特征在于,所述方法应用于参与交互的多个平台方,在隐私计算过程中计算引擎主动触发事件交互的时执行相应的方法,对每个平台方,在计算引擎的driver节点与平台之间设置有功能模块,所述功能模块集成了与平台功能相关的业务代码,所述计算引擎还包括master节点和worker节点;

所述方法包括:

在计算引擎主动触发事件交互时,由触发交互的所述master节点或所述worker节点将交互信息上报,其中,所述事件交互包括:多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知;

由功能模块根据交互信息对应的业务代码,处理对应交互行为,实现计算引擎与平台的解耦;

所述功能模块还用于设置driver节点的回调;

所述功能模块根据交互信息对应的业务代码,处理对应交互行为之前,还包括:

触发交互的所述master节点或所述worker节点将交互信息上报给driver节点;

由所述driver节点接收所述交互信息;

由所述driver节点根据所述交互信息调用所述功能模块对应交互的回调;

利用所述master节点进行计算资源的注册发现、健康检测、故障摘除、故障恢复、系统健康和系统告警功能,以及利用所述master节点进行联邦学习或者多方安全计算的任务负载均衡调度、节点间交互事件或根据分发事件进行路由分发,实现功能层面的解耦;

所述方法还包括:在计算引擎与平台之间还设置有中间件,

当计算引擎触发主动交互时,将交互信息存储至中间件,将存储在中间件的数据格式作为公开文档;

功能模块从中间件中获取对应的交互信息,解析数据格式后,处理对应的交互事件;

功能模块获取中间件中保存的数据之后,进行交互动作,并得到一个交互结果,包括:

当交互不需要返回结果的时,按照功能模块先返回交互结果至driver节点,再由driver节点返回交互结果至触发交互的master节点和worker节点;

当交互需要返回结果时,功能模块根据中间件数据格式的规范,将结果传输回中间件中,由计算引擎从中间件中获取结果,之后解析数据。

2.如权利要求1所述的方法,其特征在于,所述driver节点用于指示平台方通过开发所述功能模块来设置driver节点的回调。

3.如权利要求2所述的方法,其特征在于,所述driver节点提供了一系列规范性接口,所述规范性接口以公开文档的方式告知平台方设置回调的方法和规范。

4.如权利要求3所述的方法,其特征在于,在计算引擎主动触发事件交互之前,还:

对于各平台方,基于driver节点开发对应于平台业务代码的功能模块;

在功能模块初始化时,调用所述driver节点提供的设置回调的方法和规范,设置所述driver节点的回调。

5.如权利要求1所述的方法,其特征在于,所述功能模块还集成了计算引擎的获取注册算法、发起任务和事件通信功能。

6.如权利要求1所述的方法,其特征在于,包括:

在调用计算引擎进行计算任务时,根据开发包的开发者文档调用driver的功能,从而

调用整个计算引擎的功能,由driver定义规范的接口,给不同的平台方调用,即可实现解耦。

7.如权利要求1所述的方法,其特征在于,所述功能模块根据交互信息对应的业务代码,处理对应交互行为之前,还包括:

在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息上报;

由中间件接收所述交互信息;

所述功能模块从所述中间件获取交互信息,以通过所述功能模块执行交互的后续逻辑,实现计算引擎与平台的解耦。

8.一种计算引擎与平台解耦的系统,其特征在于,包括:

master节点和worker节点,用于在计算引擎主动触发事件交互时,触发交互的所述master节点或worker节点将交互信息上报,其中,所述事件交互包括:多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知;

功能模块,其集成了与平台功能相关的业务代码,所述功能模块用于根据交互信息对应的业务代码,处理对应交互行为,实现计算引擎与平台的解耦;

driver节点,用于接收所述master节点或worker节点上报的交互信息,并根据所述交互信息调用功能模块对应交互的回调;

利用所述master节点进行计算资源的注册发现、健康检测、故障摘除、故障恢复、系统健康和系统告警功能,以及利用所述master节点进行联邦学习或者多方安全计算的任务负载均衡调度、节点间交互事件或根据分发事件进行路由分发,实现功能层面的解耦;

所述系统还包括:在计算引擎与平台之间还设置有中间件,

当计算引擎触发主动交互时,将交互信息存储至中间件,将存储在中间件的数据格式作为公开文档;

功能模块从中间件中获取对应的交互信息,解析数据格式后,处理对应的交互事件;

功能模块获取中间件中保存的数据之后,进行交互动作,并得到一个交互结果,包括:

当交互不需要返回结果的时,按照功能模块先返回交互结果至driver节点,再由driver节点返回交互结果至触发交互的master节点和worker节点;

当交互需要返回结果时,功能模块根据中间件数据格式的规范,将结果传输回中间件中,由计算引擎从中间件中获取结果,之后解析数据。

9.如权利要求8所述的系统,其特征在于,还包括:中间件,用于存储所述交互信息;

master节点和worker节点,还用于在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息告知所述中间件;

所述功能模块还用于从所述中间件获取所述交互信息。

一种计算引擎与平台解耦的方法及系统

技术领域

[0001] 本申请涉及隐私计算领域,具体而言,涉及一种计算引擎与平台解耦的方法及系统。

背景技术

[0002] 在隐私计算领域中,各家产品计算引擎各自实现,计算引擎和平台之间相互耦合,不同厂家之间的计算引擎和平台之间很难相互集成(比如A公司的计算引擎很难被B公司的平台所集成调用),很难做到互联互通。在功能层面,计算引擎可以通过Master-worker这种主从结构,可以在计算引擎内部实现计算资源(worker,从节点)的注册发现、健康检查、故障摘除、故障恢复、系统健康、系统告警等功能,同时也可以实现联邦学习或者多方安全计算的任务负载均衡调度、节点间交互事件根据任务id进行路由分发等功能,实现功能层面的解耦。

[0003] 在调用计算引擎进行计算任务时,可以采用扩展的driver-master-worker结构,driver层实际是一个client,类似于提供一个sdk级别的开发包,平台开发人员根据开发包的开发者文档可以容易的调用driver的功能,从而调用整个计算引擎的功能,这样可以通过统一的sdk调用方式来实现不同平台调用计算引擎时的互联互通。

[0004] 在隐私计算中,还有另外一类交互,由计算引擎触发,如:1.在任务过程中,多节点间存在跨节点的事件交互,往往多方节点均处于各自的局域网内网而导致无法直连,需要通过平台中转;2.任务过程中需要在平台存储临时信息;3.在任务结束时,各方将结果存至存储后,需要将结果在存储中的file_id上报给平台;4.任务结束后,整个任务的状态需要上报至平台;5.计算资源出现各种问题,master需要往平台上报告警时等等。这部分的交互方式,不是由计算引擎定义,而是由不同厂家的平台根据自身规范所定义,因此在目前的互联互通实现上,计算引擎只能根据不同客户实现不同的交互方式,无法做到通用。

发明内容

[0005] 本申请实施例的目的在于提供一种计算引擎与平台解耦的方法及系统,用以解决对于计算引擎主动触发的交互,目前的互联互通实现方式中,计算引擎只能根据不同客户实现不同的交互方法,无法做到通用的问题。

[0006] 本申请实施例提供的一种计算引擎与平台解耦的方法,该方法应用于参与交互的多个平台方,对每个平台方,在计算引擎的driver节点与平台之间设置有功能模块,功能模块集成了与平台功能相关的业务代码,计算引擎还包括master节点和worker节点;

[0007] 一种计算引擎与平台解耦的方法,具体包括:

[0008] 在计算引擎主动触发事件交互时,由触发交互的master节点或worker节点将交互信息上报;

[0009] 由功能模块根据交互信息对应的业务代码,处理对应交互行为,实现计算引擎与平台的解耦。

[0010] 上述技术方案中,采用driver-master-worker的结构来设计计算引擎,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互信息上报,使得功能模块能够根据交互信息对应的业务代码,处理对应交互行为,即可实现计算引擎主动触发交互的解耦,解耦之后,计算引擎被集成时,由平台厂家进行适配,计算引擎不需要做适配。因此,本申请实施例的方法在互联互通的使用场景中,计算引擎可以通过通用的集成方法被集成。

[0011] 在一些可选的实施方式中,功能模块还用于设置driver节点的回调;

[0012] 功能模块根据交互信息对应的业务代码,处理对应交互行为之前,还包括:

[0013] 触发交互的master节点或worker节点将交互信息上报给driver节点;

[0014] 由driver节点接收交互信息;

[0015] 由driver节点根据交互信息调用功能模块对应交互的回调。

[0016] 上述技术方案中,采用driver-master-worker的结构来设计计算引擎,并且在driver节点中提供一系列的方法或者函数,用于设置计算引擎需要主动触发的各个交互的回调(callback)。不同厂家平台集成我方计算引擎时需要设置各个交互的回调方法或者函数,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互的内容告知driver节点,driver会调用事先设置的处理对应交互的回调方法或者函数,而回调方法或者函数的具体处理逻辑由平台方的功能模块实现,从而实现了计算引擎主动触发交互时计算引擎和平台间的解耦。

[0017] 在一些可选的实施方式中,driver节点用于指示平台方通过开发功能模块来设置driver节点的回调。

[0018] 上述技术方案中,为了实现driver调用事先设置的处理对应交互的回调方法或者函数这一过程,利用driver节点来指示平台方通过开发功能模块来设置driver节点的回调。

[0019] 在一些可选的实施方式中,driver节点的回调可以是回调方法或函数;driver节点的回调还可以是网络通信接口地址,如http url的地址,通过规范回调网络通信接口地址、发送和接收数据格式,也能达到计算引擎与平台的解耦。

[0020] 在一些可选的实施方式中,driver节点提供了一系列规范性接口,规范性接口以公开文档的方式告知平台方设置回调的方法和规范。

[0021] 上述技术方案中,driver节点中提供一系列的规范性接口,用于各个厂家平台来设置主动触发交互的回调方法或者函数。Driver节点提供的规范性接口以公开文档的方式告知各个厂家平台如何设置回调方法或者函数,以及回调方法或者函数的规范。

[0022] 在一些可选的实施方式中,在计算引擎主动触发事件交互之前,还:

[0023] 对于各平台方,基于driver节点开发对应于平台业务代码的功能模块;

[0024] 在功能模块初始化时,调用driver节点提供的设置回调的方法和规范,设置driver节点的回调。

[0025] 上述技术方案中,不同厂家根据driver节点提供的规范性文件,开发厂家自己的功能模块完成功能逻辑,功能模块内部包含2部分功能:1.平台可以通过功能模块调用计算

引擎的功能(由平台主动发起交互)。2.功能模块往driver节点中设置回调方法、函数,在计算引擎主动交互时,由driver调用功能模块设置的回调方法、函数来响应交互(由计算引擎主动发起交互)。

[0026] 在一些可选的实施方式中,功能模块还集成了计算引擎的获取注册算法、发起任务和事件通信功能。

[0027] 在一些可选的实施方式中,包括:

[0028] 利用master节点进行计算资源的注册发现、健康检测、故障摘除、故障恢复、系统健康和系统告警功能,以及利用master节点进行联邦学习或者多方安全计算的任务负载均衡调度、节点间交互事件或根据分发事件进行路由分发,实现功能层面的解耦。

[0029] 上述技术方案中,计算引擎功能上的解耦,如计算资源的注册、健康检查、故障摘除、任务调度等,这种解耦可以通过master-worker结构来实现解耦。

[0030] 在一些可选的实施方式中,包括:

[0031] 在调用计算引擎进行计算任务时,根据开发包的开发者文档调用driver的功能,从而调用整个计算引擎的功能,由driver定义规范的接口,给不同的平台方调用,即可实现解耦。

[0032] 上述技术方案中,平台调用计算引擎的解耦,如平台要发起任务、任务事件的转发等,这种解耦可以通过driver-master-worker结构,由driver定义规范的接口,给不同的平台厂商调用,即可实现解耦。

[0033] 在一些可选的实施方式中,功能模块根据交互信息对应的业务代码,处理对应交互行为之前,还包括:

[0034] 在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息上报;

[0035] 由中间件接收交互信息;

[0036] 功能模块从中间件获取交互信息,以通过功能模块执行交互的后续逻辑,实现计算引擎与平台的解耦。

[0037] 上述技术方案中,在部署时,增加中间件模块,如消息队列(MQ)等。在安全等级较高的场景下,比如银行或者政务系统的厂家,可以采用厂家自研的消息中间件。当计算引擎触发主动交互时,可以将交互信息存储至中间件,将存储在中间件的数据格式作为公开文档。不同厂家在实现功能模块时,可以从中间件中获取对应的交互信息,解析后,可以利用功能模块进行后续的功能逻辑实现。这种方式不通过driver对功能模块的回调机制,同样可以实现计算引擎在主动触发交互时计算引擎和平台之间的解耦。其中,不同厂家根据driver节点提供的规范性文件和依赖的中间件的数据格式的规范性文件,开发厂家自己的功能模块完成功能逻辑,功能模块内部包含2部分功能:1.平台可以通过功能模块调用计算引擎的功能(由平台主动发起交互)。2.在计算引擎主动交互时,功能模块获取中间件中保存的数据,解析数据格式,随后处理对应的交互事件(由计算引擎主动发起交互)。

[0038] 本申请实施例提供的一种计算引擎与平台解耦的系统,包括:

[0039] master节点和worker节点,用于在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息上报;

[0040] 功能模块,其集成了与平台功能相关的业务代码,功能模块用于根据交互信息对

应的业务代码,处理对应交互行为,实现计算引擎与平台的解耦。

[0041] 上述技术方案中,采用driver-master-worker的结构来设计计算引擎,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互信息上报,使得功能模块能够根据交互信息对应的业务代码,处理对应交互行为,即可实现计算引擎主动触发交互的解耦,解耦之后,计算引擎被集成时,由平台厂家进行适配,计算引擎不需要做适配。因此,本申请实施例的方法在互联互通的使用场景中,计算引擎可以通过通用的集成方法被集成。

[0042] 在一些可选的实施方式中,还包括:

[0043] driver节点,用于接收master节点或worker节点上报的交互信息,并根据交互信息调用功能模块对应交互的回调。

[0044] 上述技术方案中,采用driver-master-worker的结构来设计计算引擎,并且在driver节点中提供一系列的方法或者函数,用于设置计算引擎需要主动触发的各个交互的回调(callback)。不同厂家平台集成我方计算引擎时需要设置各个交互的回调方法或者函数,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互的内容告知driver节点,driver会调用事先设置的处理对应交互的回调方法或者函数,而回调方法或者函数的具体处理逻辑由平台方的功能模块实现,从而实现了计算引擎主动触发交互时计算引擎和平台间的解耦。

[0045] 在一些可选的实施方式中,还包括:中间件,用于存储交互信息;

[0046] master节点和worker节点,还用于在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息告知中间件;

[0047] 功能模块还用于从中间件获取交互信息。

[0048] 上述技术方案中,在部署时,增加中间件模块,如消息队列(MQ)等。在安全等级较高的场景下,比如银行或者政务系统的厂家,可以采用厂家自研的消息中间件。当计算引擎触发主动交互时,可以将交互信息存储至中间件,将存储在中间件的数据格式作为公开文档。不同厂家在实现功能模块时,可以从中间件中获取对应的交互信息,解析后,可以利用功能模块进行后续的功能逻辑实现。这种方式不通过driver对功能模块的回调机制,同样可以实现计算引擎在主动触发交互时计算引擎和平台之间的解耦。其中,不同厂家根据driver节点提供的规范性文件和依赖的中间件的数据格式的规范性文件,开发厂家自己的功能模块完成功能逻辑,功能模块内部包含2部分功能:1.平台可以通过功能模块调用计算引擎的功能(由平台主动发起交互)。2.在计算引擎主动交互时,功能模块获取中间件中保存的数据,解析数据格式,随后处理对应的交互事件(由计算引擎主动发起交互)。

附图说明

[0049] 为了更清楚地说明本申请实施例的技术方案,下面将对本申请实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本申请的某些实施例,因此不应被看作是对范围的限定,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他相关的附图。

[0050] 图1为本申请实施例提供的一种计算引擎与平台解耦的方法步骤流程图;

[0051] 图2为本申请实施例提供的一种计算引擎与平台解耦的系统的结构示意图;

[0052] 图3为本申请实施例中计算引擎与平台解耦的系统的工作流程示意图;

[0053] 图4为本申请另一实施例提供的计算引擎与平台解耦的系统的结构示意图。

具体实施方式

[0054] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行描述。

[0055] 本申请实施例提供的一种计算引擎与平台解耦的方法,该方法应用于参与交互的多个平台方,对每个平台方,在计算引擎的driver节点与平台之间设置有功能模块,功能模块集成了与平台功能相关的业务代码,计算引擎还包括master节点和worker节点。

[0056] 请参照图1,图1为一种计算引擎与平台解耦的方法步骤流程图,具体包括:

[0057] 步骤100、在计算引擎主动触发事件交互时,由触发交互的master节点或worker节点将交互信息上报;

[0058] 步骤200、由功能模块根据交互信息对应的业务代码,处理对应交互行为,实现计算引擎与平台的解耦。

[0059] 本申请实施例中,采用driver-master-worker的结构来设计计算引擎,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互信息上报,使得功能模块能够根据交互信息对应的业务代码,处理对应交互行为,即可实现计算引擎主动触发交互的解耦,解耦之后,计算引擎被集成时,由平台厂家进行适配,计算引擎不需要做适配。因此,本申请实施例的方法在互联互通的使用场景中,计算引擎可以通过通用的集成方法被集成。

[0060] 在一些可选的实施方式中,功能模块还用于设置driver节点的回调;功能模块根据交互信息对应的业务代码,处理对应交互行为之前,还包括步骤101-步骤103:

[0061] 步骤101、触发交互的master节点或worker节点将交互信息上报给driver节点;

[0062] 步骤102、由driver节点接收交互信息;

[0063] 步骤103、由driver节点根据交互信息调用功能模块对应交互的回调。

[0064] 本申请实施例中,采用driver-master-worker的结构来设计计算引擎,并且在driver节点中提供一系列的方法或者函数,用于设置计算引擎需要主动触发的各个交互的回调(callback)。不同厂家平台集成我方计算引擎时需要设置各个交互的回调方法或者函数,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互的内容告知driver节点,driver会调用事先设置的处理对应交互的回调方法或者函数,而回调方法或者函数的具体处理逻辑由平台方的功能模块实现,从而实现了计算引擎主动触发交互时计算引擎和平台间的解耦。

[0065] 在一些可选的实施方式中,driver节点用于指示平台方通过开发功能模块来设置driver节点的回调。

[0066] 本申请实施例中,为了实现driver调用事先设置的处理对应交互的回调方法或者函数这一过程,利用driver节点来指示平台方通过开发功能模块来设置driver节点的回

调。

[0067] 在一些可选的实施方式中,driver节点的回调可以是回调方法或函数;driver节点的回调还可以是网络通信接口地址,如http url的地址,通过规范回调网络通信接口地址、发送和接收数据格式,也能达到计算引擎与平台的解耦。

[0068] 在一些可选的实施方式中,driver节点提供了一系列规范性接口,规范性接口以公开文档的方式告知平台方设置回调的方法和规范。

[0069] 本申请实施例中,driver节点中提供一系列的规范性接口,用于各个厂家平台来设置主动触发交互的回调方法或者函数。Driver节点提供的规范性接口以公开文档的方式告知各个厂家平台如何设置回调方法或者函数,以及回调方法或者函数的规范。

[0070] 在一些可选的实施方式中,在计算引擎主动触发事件交互之前,还:

[0071] 对于各平台方,基于driver节点开发对应于平台业务代码的功能模块;

[0072] 在功能模块初始化时,调用driver节点提供的设置回调的方法和规范,设置driver节点的回调。

[0073] 本申请实施例中,不同厂家根据driver节点提供的规范性文件,开发厂家自己的功能模块完成功能逻辑,功能模块内部包含2部分功能:1.平台可以通过功能模块调用计算引擎的功能(由平台主动发起交互)。2.功能模块往driver节点中设置回调方法、函数,在计算引擎主动交互时,由driver调用功能模块设置的回调方法、函数来响应交互(由计算引擎主动发起交互)。

[0074] 在一些可选的实施方式中,功能模块还集成了计算引擎的获取注册算法、发起任务和事件通信功能。

[0075] 在一些可选的实施方式中,包括:

[0076] 利用master节点进行计算资源的注册发现、健康检测、故障摘除、故障恢复、系统健康和系统告警功能,以及利用master节点进行联邦学习或者多方安全计算的任务负载均衡调度、节点间交互事件或根据分发事件进行路由分发,实现功能层面的解耦。

[0077] 本申请实施例中,计算引擎功能上的解耦,如计算资源的注册、健康检查、故障摘除、任务调度等,这种解耦可以通过master-worker结构来实现解耦。

[0078] 在一些可选的实施方式中,包括:

[0079] 在调用计算引擎进行计算任务时,根据开发包的开发者文档调用driver的功能,从而调用整个计算引擎的功能,由driver定义规范的接口,给不同的平台方调用,即可实现解耦。

[0080] 本申请实施例中,平台调用计算引擎的解耦,如平台要发起任务、任务事件的转发等,这种解耦可以通过driver-master-worker结构,由driver定义规范的接口,给不同的平台厂商调用,即可实现解耦。

[0081] 在一些可选的实施方式中,功能模块根据交互信息对应的业务代码,处理对应交互行为之前,还包括步骤111-步骤113:

[0082] 步骤111、在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息上报;

[0083] 步骤112、由中间件接收交互信息;

[0084] 步骤113、功能模块从中间件获取交互信息,以通过功能模块执行交互的后续逻

辑,实现计算引擎与平台的解耦。

[0085] 本申请实施例中,在部署时,增加中间件模块,如消息队列(MQ)等。在安全等级较高的场景下,比如银行或者政务系统的厂家,可以采用厂家自研的消息中间件。当计算引擎触发主动交互时,可以将交互信息存储至中间件,将存储在中间件的数据格式作为公开文档。不同厂家在实现功能模块时,可以从中间件中获取对应的交互信息,解析后,可以利用功能模块进行后续的功能逻辑实现。这种方式不通过driver对功能模块的回调机制,同样可以实现计算引擎在主动触发交互时计算引擎和平台之间的解耦。其中,不同厂家根据driver节点提供的规范性文件和依赖的中间件的数据格式的规范性文件,开发厂家自己的功能模块完成功能逻辑,功能模块内部包含2部分功能:1.平台可以通过功能模块调用计算引擎的功能(由平台主动发起交互)。2.在计算引擎主动交互时,功能模块获取中间件中保存的数据,解析数据格式,随后处理对应的交互事件(由计算引擎主动发起交互)。

[0086] 请参照图2,图2为本申请实施例提供的一种计算引擎与平台解耦的系统的结构示意图,包括:

[0087] master节点和worker节点,用于在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息上报;

[0088] 功能模块,其集成了与平台功能相关的业务代码,功能模块用于根据交互信息对应的业务代码,处理对应交互行为,实现计算引擎与平台的解耦。

[0089] 本申请实施例中,采用driver-master-worker的结构来设计计算引擎,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互信息上报,使得功能模块能够根据交互信息对应的业务代码,处理对应交互行为,即可实现计算引擎主动触发交互的解耦,解耦之后,计算引擎被集成时,由平台厂家进行适配,计算引擎不需要做适配。因此,本申请实施例的方法在互联互通的使用场景中,计算引擎可以通过通用的集成方法被集成。

[0090] 在一些可选的实施方式中,还包括:driver节点,用于接收master节点或worker节点上报的交互信息,并根据交互信息调用功能模块对应交互的回调。

[0091] 本申请实施例中,采用driver-master-worker的结构来设计计算引擎,并且在driver节点中提供一系列的方法或者函数,用于设置计算引擎需要主动触发的各个交互的回调(callback)。不同厂家平台集成我方计算引擎时需要设置各个交互的回调方法或者函数,当计算引擎触发一个交互时(比如多节点间事件交互、临时数据存储、结果上报、状态上报、告警通知等),这时对应的触发交互的节点(worker或者master)会将该交互的内容告知driver节点,driver会调用事先设置的处理对应交互的回调方法或者函数,而回调方法或者函数的具体处理逻辑由平台方的功能模块实现,从而实现了计算引擎主动触发交互时计算引擎和平台间的解耦。

[0092] 请参照图3,图3为本申请实施例中计算引擎与平台解耦的系统的工作流程示意图,具体包括:

[0093] 在计算引擎主动触发事件交互之前,利用driver节点提供一系列规范性接口,所述规范性接口以公开文档的方式告知平台方设置回调的方法和规范。对于各平台方,基于driver节点开发对应于平台业务代码的功能模块,并在功能模块初始化时,调用所述

driver节点提供的设置回调的方法和规范,设置所述driver节点的回调。

[0094] 在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息上报给driver节点;由driver节点接收交互信息;由driver节点根据交互信息调用功能模块对应交互的回调。

[0095] 之后,功能模块根据交互信息对应的业务代码,处理对应交互行为,实现计算引擎与平台的解耦。

[0096] 最后,将交互结构原链路返回,及功能模块的交互结果返回给driver节点,driver节点再返回至触发交互的master节点或worker节点。

[0097] 在一些可选的实施方式中,请参照图4,图4为本申请另一实施例提供的计算引擎与平台解耦的系统的结构示意图,该系统还包括:中间件,用于存储交互信息;master节点和worker节点,还用于在计算引擎主动触发事件交互时,触发交互的master节点或worker节点将交互信息告知中间件;功能模块还用于从中间件获取交互信息。

[0098] 本申请实施例中,在部署时,增加中间件模块,如消息队列(MQ)等。在安全等级较高的场景下,比如银行或者政务系统的厂家,可以采用厂家自研的消息中间件。当计算引擎触发主动交互时,可以将交互信息存储至中间件,将存储在中间件的数据格式作为公开文档。不同厂家在实现功能模块时,可以从中间件中获取对应的交互信息,解析后,可以利用功能模块进行后续的功能逻辑实现。这种方式不通过driver对功能模块的回调机制,同样可以实现计算引擎在主动触发交互时计算引擎和平台之间的解耦。其中,不同厂家根据driver节点提供的规范性文件和依赖的中间件的数据格式的规范性文件,开发厂家自己的功能模块完成功能逻辑,功能模块内部包含2部分功能:1.平台可以通过功能模块调用计算引擎的功能(由平台主动发起交互)。2.在计算引擎主动交互时,功能模块获取中间件中保存的数据,解析数据格式,随后处理对应的交互事件(由计算引擎主动发起交互)。

[0099] 同样的,功能模块获取中间件中保存的数据之后,进行交互动作,并得到一个交互结果,由于中间件模块的特殊性,分为2种情况:

[0100] 第一种情况是交互不需要返回结果的情况下,还是按照功能模块先返回交互结果至driver节点,再由driver节点返回交互结果至触发交互的master节点和worker节点。

[0101] 第二种情况是交互需要返回结果的情况下,功能模块需要根据中间件数据格式的规范,将结果传输回中间件中,由计算引擎从中间件中获取获取结果,之后解析数据。

[0102] 在本申请所提供的实施例中,应该理解到,所揭露装置和方法,可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,又例如,多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些通信接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0103] 另外,作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0104] 再者,在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部

分,也可以是各个模块单独存在,也可以两个或两个以上模块集成形成一个独立的部分。

[0105] 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。

[0106] 以上所述仅为本申请的实施例而已,并不用于限制本申请的保护范围,对于本领域的技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。

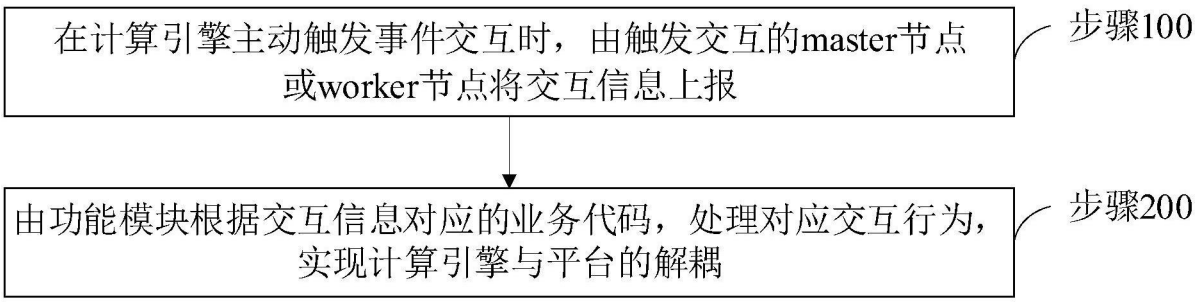

在计算引擎主动触发事件交互时，由触发交互的master节点或worker节点将交互信息上报
步骤100
由功能模块根据交互信息对应的业务代码，处理对应交互行为，实现计算引擎与平台的解耦
步骤200

图1

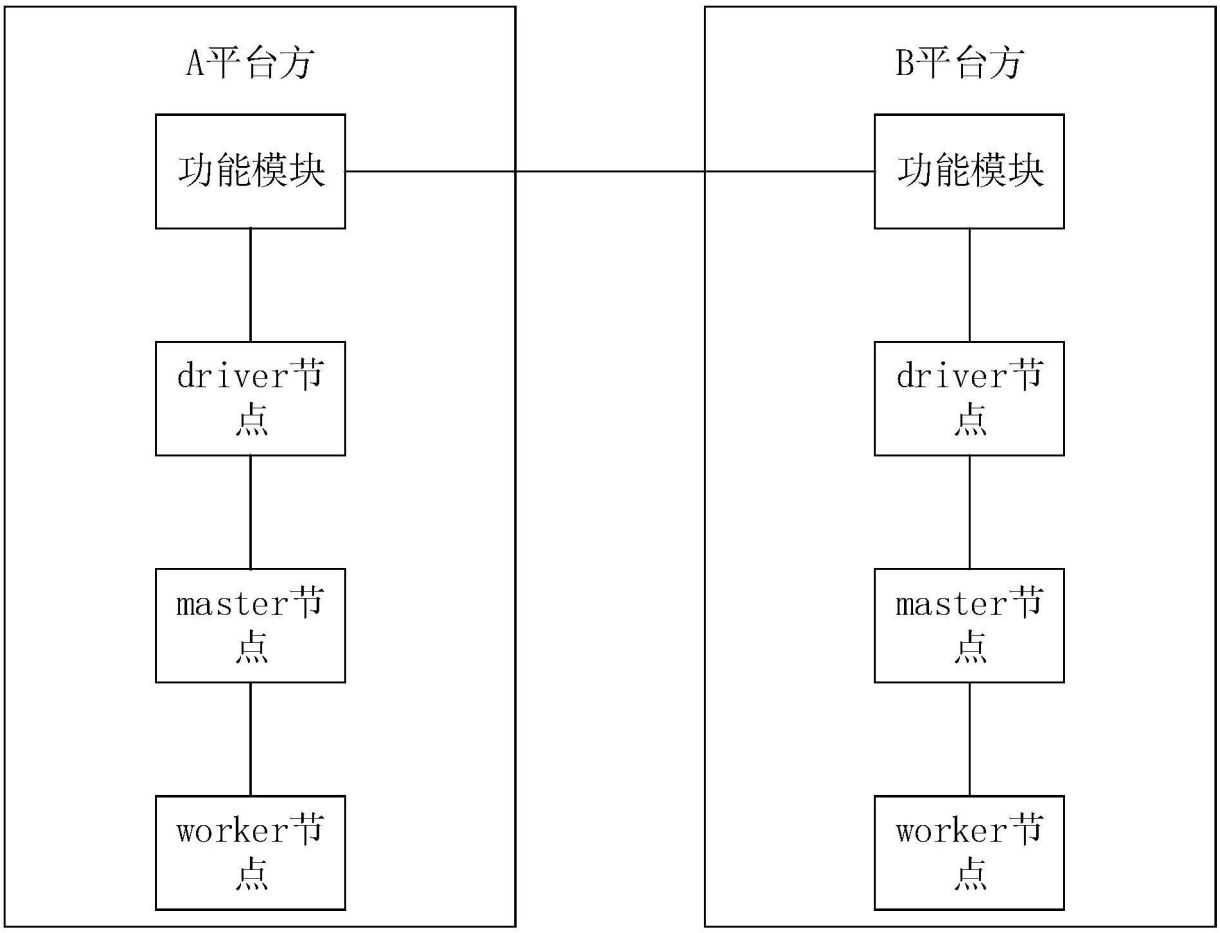

A平台方
功能模块
driver节点
master节点
worker节点
B平台方
功能模块
driver节点
master节点
worker节点

图2

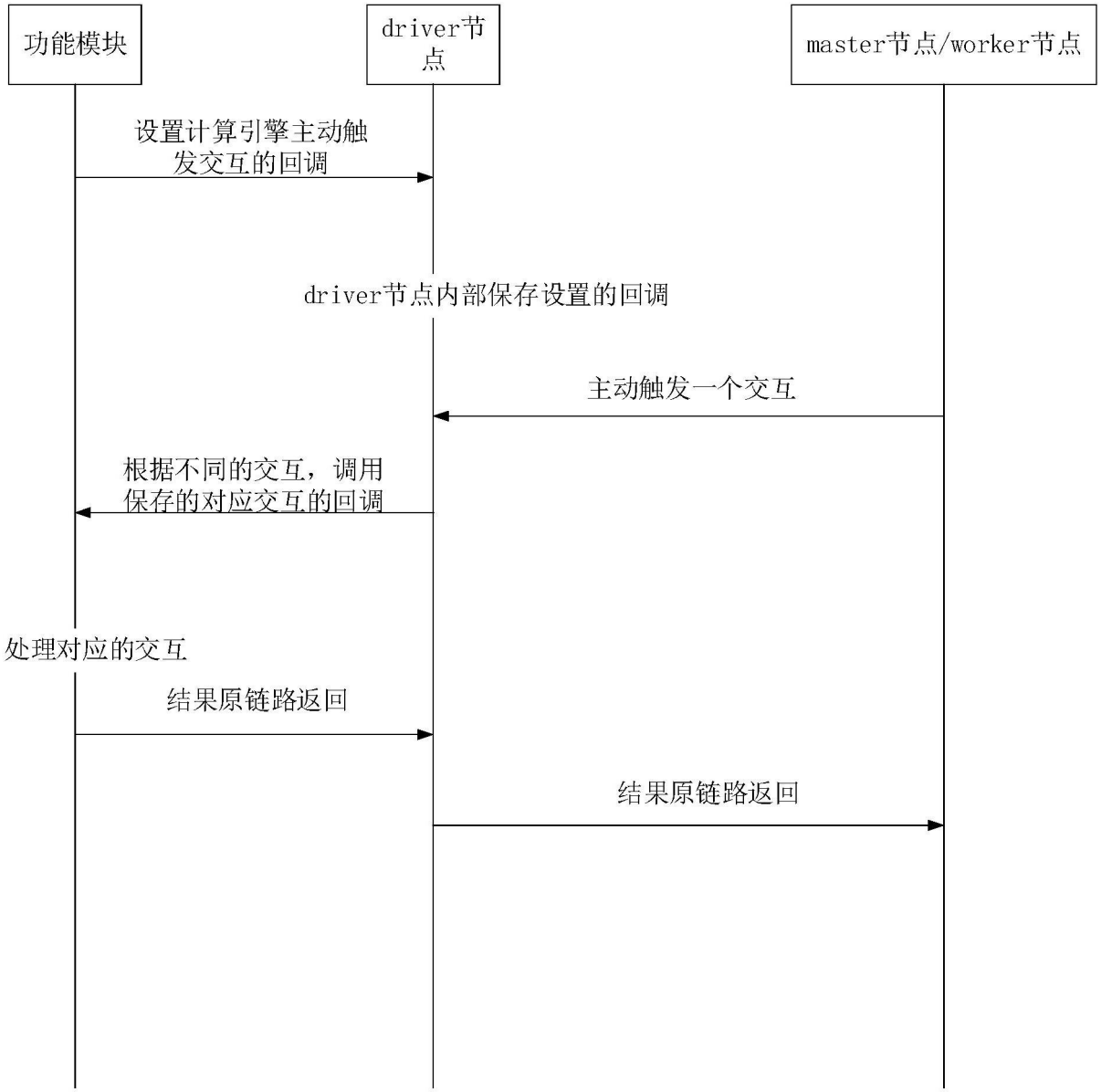

功能模块
driver节点
master节点/worker节点
设置计算引擎主动触发交互的回调
driver节点内部保存设置的回调
主动触发一个交互
根据不同的交互，调用保存的对应交互的回调
处理对应的交互
结果原链路返回
结果原链路返回

图3

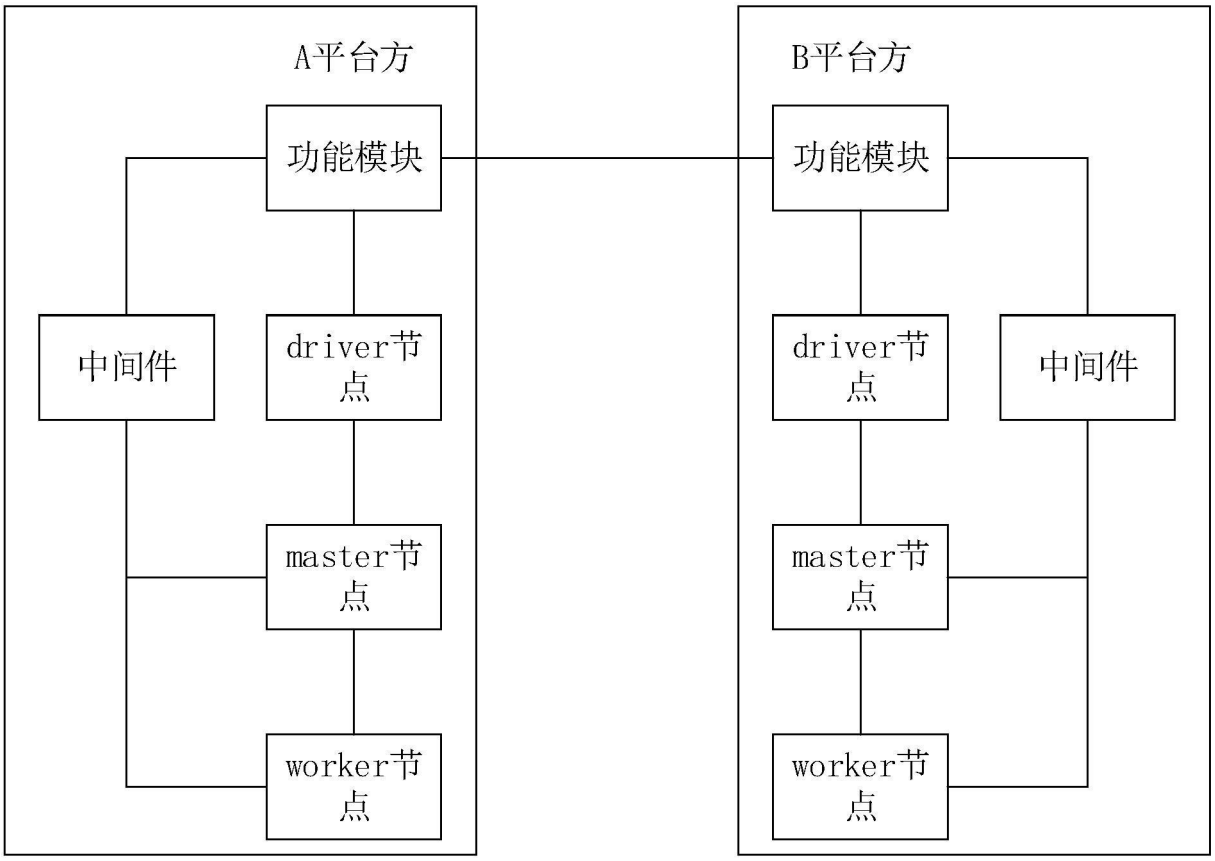

A平台方
功能模块
中间件
driver节点
master节点
worker节点
B平台方
功能模块
driver节点
中间件
master节点
worker节点

图4