(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 117632443 B
(45)授权公告日 2024.04.26

(21)申请号 202410105221.9
(22)申请日 2024.01.25
(65)同一申请的已公布的文献号
申请公布号 CN 117632443 A
(43)申请公布日 2024.03.01
(73)专利权人 腾讯科技(深圳)有限公司
地址 518031 广东省深圳市南山区高新区科技中一路腾讯大厦35层
(72)发明人 史高雄 蔡明师 林万鹏 王镇
(74)专利代理机构 深圳市联鼎知识产权代理有限公司 44232
专利代理师 徐明霞
(51)Int.Cl.
*G06F 9/48*(2006.01)
*G06Q 10/0633*(2023.01)
*G06Q 10/10*(2023.01)

(56)对比文件
CN 114138528 A,2022.03.04
CN 111768288 A,2020.10.13
CN 106776127 A,2017.05.31
JP 2005128658 A,2005.05.19
US 9904899 B2,2018.02.27
CN 117056116 A,2023.11.14

审查员 赵静

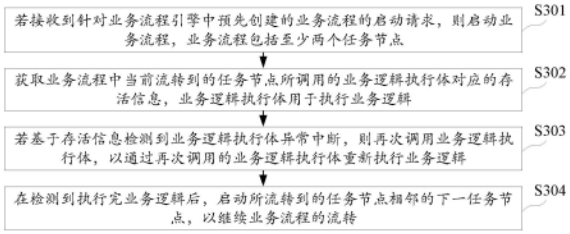
权利要求书4页 说明书19页 附图6页

(54)发明名称

# 业务流程的流转控制方法及装置、设备、介质

(57)摘要

本申请的实施例公开了一种业务流程的流转控制方法及装置、设备、介质,可以应用于智慧交通、辅助驾驶、云技术、人工智能等场景中。该方法包括:若接收到针对业务流程引擎中预先创建的业务流程的启动请求,则启动业务流程,业务流程包括至少两个任务节点;获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,业务逻辑执行体用于执行业务逻辑;若基于存活信息检测到业务逻辑执行体异常中断,则再次调用业务逻辑执行体,以通过再次调用的业务逻辑执行体重新执行业务逻辑;在检测到执行完业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,以继续业务流程的流转。本申请的技术方案提升了业务流程的流转控制可靠性。




CN 117632443 B

1.一种业务流程的流转控制方法,其特征在于,包括:

若接收到针对业务流程引擎中预先创建的业务流程的启动请求,则启动所述业务流程,所述业务流程包括至少两个任务节点;

获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,所述业务逻辑执行体用于执行业务逻辑;

若基于所述存活信息检测到所述业务逻辑执行体异常中断,则再次调用所述业务逻辑执行体,以通过再次调用的业务逻辑执行体重新执行所述业务逻辑;

在检测到执行完所述业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,以继续所述业务流程的流转;

其中,所述获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,包括:

获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的标识信息;

从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息;其中,所述指定存储区域中存储的所述业务逻辑执行体的存活信息是所述业务逻辑执行体在执行业务逻辑过程中持续生成并发送的;

其中,所述存活信息包括每个心跳消息发送的时间戳信息,所述心跳消息包括所述业务逻辑执行体的标识信息和所述业务逻辑执行体所属的业务流程的标识信息;

所述从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息,包括:

从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合;其中,同一存活信息集合包括同一业务逻辑执行体所针对的同一业务流程对应的多个时间戳信息。

2.根据权利要求1所述的方法,其特征在于,在所述从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合之后,所述方法还包括:

计算当前时间点与所选择出的存活信息集合中多个时间戳信息之间的间隔时长;

基于所计算得到的间隔时长与预设间隔时长阈值之间的关系,检测所述业务逻辑执行体的存活情况。

3.根据权利要求2所述的方法,其特征在于,所述计算当前时间点与所选择出的存活信息集合中多个时间戳信息之间的间隔时长,包括:

从所选择出的存活信息集合的多个时间戳信息中选择距离当前时间点最近的时间戳信息,并计算所选择出的时间戳信息与所述当前时间点之间的间隔时长;或者

计算所述当前时间点与所选择出的存活信息集合的每个时间戳信息之间的间隔时长,得到多个间隔时长,并从所述多个间隔时长中选择间隔时长最小的间隔时长。

4.根据权利要求2所述的方法,其特征在于,所述基于所计算得到的间隔时长与预设间隔时长阈值之间的关系,检测所述业务逻辑执行体的存活情况,包括:

若所计算得到的间隔时长大于预设间隔时长阈值,则得到用于表征所述业务逻辑执行体异常中断的检测结果;

若所计算得到的间隔时长等于或小于所述预设间隔时长阈值,则得到用于表征所述业务逻辑执行体正常执行的检测结果。

5.根据权利要求1所述的方法,其特征在于,所述方法还包括:

接收所述业务逻辑执行体发送的执行完成通知信息;其中,所述执行完成通知信息是在所述业务逻辑执行体执行完业务逻辑后生成的;

基于所述执行完成通知信息从所述指定存储区域中删除所述业务逻辑执行体的存活信息。

6.根据权利要求1至5中任一项所述的方法,其特征在于,所述获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,包括:

启动业务逻辑执行体检测服务;

通过所启动的业务逻辑执行体检测服务周期性获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,并基于所获取到的存活信息检测所述业务逻辑执行体的存活情况。

7.根据权利要求1至5中任一项所述的方法,其特征在于,所述再次调用所述业务逻辑执行体,包括:

调用所述业务流程引擎的业务重试接口;

通过所述业务重试接口重新启动所流转到的任务节点,以基于所重新启动的任务节点再次调用所述业务逻辑执行体。

8.根据权利要求1至5中任一项所述的方法,其特征在于,所述方法还包括:

若接收到软中断请求,则获取当前处于执行状态的业务逻辑执行体;

等待所述处于执行状态的业务逻辑执行体执行完业务逻辑;

在所述处于执行状态的业务逻辑执行体执行完业务逻辑后,控制退出所述业务流程。

9.根据权利要求8所述的方法,其特征在于,所述获取当前处于执行状态的业务逻辑执行体,包括:

获取用于记录处于执行状态的业务逻辑执行体的数量的数量参数;其中,所述数量参数是通过在每调用一个业务逻辑执行体时进行指定单位量的增加,以及在每一个业务逻辑执行体执行完业务逻辑时进行所述指定单位量的减少所得到的;

若检测到所述数量参数所表征的处于执行状态的业务逻辑执行体的数量为至少一个,则确定当前存在处于执行状态的业务逻辑执行体;

所述在所述处于执行状态的业务逻辑执行体执行完业务逻辑后,控制退出所述业务流程,包括:

若检测到所述数量参数所表征的处于执行状态的业务逻辑执行体的数量为零,则控制退出所述业务流程。

10.根据权利要求8所述的方法,其特征在于,所述方法还包括:

若检测到触发等待所述处于执行状态的业务逻辑执行体执行完业务逻辑,则记录等待的开始时间点,以及从所述开始时间点进行计时;

若检测到计时时长达到预设计时时长阈值,则控制退出所述业务流程。

11.一种业务流程的流转控制装置,其特征在于,包括:

第一启动模块,配置为若接收到针对业务流程引擎中预先创建的业务流程的启动请求,则启动所述业务流程,所述业务流程包括至少两个任务节点;

获取模块,配置为获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,所述业务逻辑执行体用于执行业务逻辑;

调用模块,配置为若基于所述存活信息检测到所述业务逻辑执行体异常中断,则再次调用所述业务逻辑执行体,以通过再次调用的业务逻辑执行体重新执行所述业务逻辑;

第二启动模块,配置为在检测到执行完所述业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,以继续所述业务流程的流转;

其中,所述获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,包括:

获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的标识信息;

从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息;其中,所述指定存储区域中存储的所述业务逻辑执行体的存活信息是所述业务逻辑执行体在执行业务逻辑过程中持续生成并发送的;

其中,所述存活信息包括每个心跳消息发送的时间戳信息,所述心跳消息包括所述业务逻辑执行体的标识信息和所述业务逻辑执行体所属的业务流程的标识信息;

所述从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息,包括:

从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合;其中,同一存活信息集合包括同一业务逻辑执行体所针对的同一业务流程对应的多个时间戳信息。

12.根据权利要求11所述的装置,其特征在于,所述装置还包括:

计算模块,配置为计算当前时间点与所选择出的存活信息集合中多个时间戳信息之间的间隔时长;

检测模块,配置为基于所计算得到的间隔时长与预设间隔时长阈值之间的关系,检测所述业务逻辑执行体的存活情况。

13.根据权利要求12所述的装置,其特征在于,所述计算模块,具体配置为:

从所选择出的存活信息集合的多个时间戳信息中选择距离当前时间点最近的时间戳信息,并计算所选择出的时间戳信息与所述当前时间点之间的间隔时长;或者

计算所述当前时间点与所选择出的存活信息集合的每个时间戳信息之间的间隔时长,得到多个间隔时长,并从所述多个间隔时长中选择间隔时长最小的间隔时长。

14.一种电子设备,其特征在于,包括:

一个或多个处理器;

存储器,用于存储一个或多个程序,当所述一个或多个程序被所述电子设备执行时,使得所述电子设备实现如权利要求1至10中任一项所述的业务流程的流转控制方法。

15.一种计算机可读介质,其上存储有计算机程序,其特征在于,所述计算机程序被处

理器执行时实现如权利要求1至10中任一项所述的业务流程的流转控制方法。

16.一种计算机程序产品,包括计算机指令,其特征在于,所述计算机指令被处理器执行时实现如权利要求1至10中任一项所述的业务流程的流转控制方法。

# 业务流程的流转控制方法及装置、设备、介质

## 技术领域

[0001] 本申请涉及计算机技术领域,具体而言,涉及一种业务流程的流转控制方法、业务流程的流转控制装置、电子设备、计算机可读介质。

## 背景技术

[0002] 目前,通过业务终端执行某一业务流程来实现某种业务目的是常见业务处理方式;其中,业务流程是将多个任务节点按照业务逻辑连接起来确定的。相关技术在业务流程的流转过程中,容易出现任务节点的任务驱动中断,造成业务流程执行失败的现象。

[0003] 因此,如何提升业务流程的流转控制可靠性是亟待解决的问题。

## 发明内容

[0004] 本申请的实施例提供了一种业务流程的流转控制方法及装置、设备、介质,提升了业务流程的流转控制可靠性。

[0005] 第一方面,本申请实施例提供了一种业务流程的流转控制方法,所述方法包括:若接收到针对业务流程引擎中预先创建的业务流程的启动请求,则启动所述业务流程,所述业务流程包括至少两个任务节点;获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,所述业务逻辑执行体用于执行业务逻辑;若基于所述存活信息检测到所述业务逻辑执行体异常中断,则再次调用所述业务逻辑执行体,以通过再次调用的业务逻辑执行体重新执行所述业务逻辑;在检测到执行完所述业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,以继续所述业务流程的流转。

[0006] 第二方面,本申请实施例提供了一种业务流程的流转控制装置,所述装置包括:第一启动模块,配置为若接收到针对业务流程引擎中预先创建的业务流程的启动请求,则启动所述业务流程,所述业务流程包括至少两个任务节点;获取模块,配置为获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,所述业务逻辑执行体用于执行业务逻辑;调用模块,配置为若基于所述存活信息检测到所述业务逻辑执行体异常中断,则再次调用所述业务逻辑执行体,以通过再次调用的业务逻辑执行体重新执行所述业务逻辑;第二启动模块,配置为在检测到执行完所述业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,以继续所述业务流程的流转。

[0007] 在本申请的一个实施例中,基于前述方案,所述获取模块,具体配置为:获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的标识信息;从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息;其中,所述指定存储区域中存储的所述业务逻辑执行体的存活信息是所述业务逻辑执行体在执行业务逻辑过程中持续生成并发送的。

[0008] 在本申请的一个实施例中,基于前述方案,所述存活信息包括每个心跳消息发送的时间戳信息,所述心跳消息包括所述业务逻辑执行体的标识信息和所述业务逻辑执行体所属的业务流程的标识信息;所述获取模块,还具体配置为:从指定存储区域所存储的多个

存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合;其中,同一存活信息集合包括同一业务逻辑执行体所针对的同一业务流程对应的多个时间戳信息。

[0009] 在本申请的一个实施例中,基于前述方案,在所述从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合之后,所述装置还包括:计算模块,配置为计算当前时间点与所选择出的存活信息集合中多个时间戳信息之间的间隔时长;检测模块,配置为基于所计算得到的间隔时长与预设间隔时长阈值之间的关系,检测所述业务逻辑执行体的存活情况。

[0010] 在本申请的一个实施例中,基于前述方案,所述计算模块,具体配置为:从所选择出的存活信息集合的多个时间戳信息中选择距离当前时间点最近的时间戳信息,并计算所选择出的时间戳信息与所述当前时间点之间的间隔时长;或者计算所述当前时间点与所选择出的存活信息集合的每个时间戳信息之间的间隔时长,得到多个间隔时长,并从所述多个间隔时长中选择间隔时长最小的间隔时长。

[0011] 在本申请的一个实施例中,基于前述方案,所述检测模块,具体配置为:若所计算得到的间隔时长大于预设间隔时长阈值,则得到用于表征所述业务逻辑执行体异常中断的检测结果;若所计算得到的间隔时长等于或小于所述预设间隔时长阈值,则得到用于表征所述业务逻辑执行体正常执行的检测结果。

[0012] 在本申请的一个实施例中,基于前述方案,所述装置还包括:接收模块,配置为接收所述业务逻辑执行体发送的执行完成通知信息;其中,所述执行完成通知信息是在所述业务逻辑执行体执行完业务逻辑后生成的;删除模块,配置为基于所述执行完成通知信息从所述指定存储区域中删除所述业务逻辑执行体的存活信息。

[0013] 在本申请的一个实施例中,基于前述方案,所述获取模块,具体配置为:启动业务逻辑执行体检测服务;通过所启动的业务逻辑执行体检测服务周期性获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,并基于所获取到的存活信息检测所述业务逻辑执行体的存活情况。

[0014] 在本申请的一个实施例中,基于前述方案,所述调用模块,具体配置为:调用所述业务流程引擎的业务重试接口;通过所述业务重试接口重新启动所流转到的任务节点,以基于所重新启动的任务节点再次调用所述业务逻辑执行体。

[0015] 在本申请的一个实施例中,基于前述方案,所述装置还包括退出控制模块,配置为:若接收到软中断请求,则获取当前处于执行状态的业务逻辑执行体;等待所述处于执行状态的业务逻辑执行体执行完业务逻辑;在所述处于执行状态的业务逻辑执行体执行完业务逻辑后,控制退出所述业务流程。

[0016] 在本申请的一个实施例中,基于前述方案,所述退出控制模块,具体配置为:获取用于记录处于执行状态的业务逻辑执行体的数量的数量参数;其中,所述数量参数是通过在每调用一个业务逻辑执行体时进行指定单位量的增加,以及在每一个业务逻辑执行体执行完业务逻辑时进行所述指定单位量的减少所得到的;若检测到所述数量参数所表征的处于执行状态的业务逻辑执行体的数量为至少一个,则确定当前存在处于执行状态的业务逻辑执行体;若检测到所述数量参数所表征的处于执行状态的业务逻辑执行体的数量为零,

则控制退出所述业务流程。

[0017] 在本申请的一个实施例中,基于前述方案,所述退出控制模块,还具体配置为:若检测到触发等待所述处于执行状态的业务逻辑执行体执行完业务逻辑,则记录等待的开始时间点,以及从所述开始时间点进行计时;若检测到计时时长达到预设计时时长阈值,则控制退出所述业务流程。

[0018] 第三方面,本申请实施例提供了一种电子设备,包括一个或多个处理器;存储器,用于存储一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行时,使得所述电子设备实现如上所述的业务流程的流转控制方法。

[0019] 第四方面,本申请实施例提供了一种计算机可读介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现如上所述的业务流程的流转控制方法。

[0020] 第五方面,本申请实施例提供了一种计算机程序产品,包括计算机指令,所述计算机指令被处理器执行时实现如上所述的业务流程的流转控制方法。

[0021] 在本申请的实施例提供的技术方案中:

[0022] 在启动含有任务节点的业务流程后,会获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,并在基于存活信息检测到业务逻辑执行体异常中断后,触发业务逻辑执行体的再次调用,以通过再次调用的业务逻辑执行体重新执行业务逻辑,同时在检测到执行完业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,由此实现业务流程的流转控制。

[0023] 也即,通过业务逻辑执行体的存活信息实时检测业务逻辑执行体是否发生异常中断,并在其异常中断后对其进行再次调用,避免了业务逻辑执行体异常中断造成业务流程执行失败的现象,保证了业务流程的正常执行/流转,提升了业务流程的流转控制准确性、灵活性,业务流程的流转控制可靠性高。

[0024] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本申请。

## 附图说明

[0025] 图1是本申请的一示例性实施例示出的业务流程执行失败的示意图。

[0026] 图2是可以应用本申请实施例的技术方案的示例性实施环境的示意图。

[0027] 图3是本申请的一示例性实施例示出的业务流程的流转控制方法的流程图。

[0028] 图4是本申请的另一示例性实施例示出的业务流程的流转控制方法的流程图。

[0029] 图5是本申请的另一示例性实施例示出的业务流程的流转控制方法的流程图。

[0030] 图6是本申请的另一示例性实施例示出的业务流程的流转控制方法的示意图。

[0031] 图7是本申请的另一示例性实施例示出的业务流程的流转控制方法的流程图。

[0032] 图8是本申请的另一示例性实施例示出的业务流程的流转控制方法的流程图。

[0033] 图9是本申请的一示例性实施例示出的业务流程的流转控制装置的框图。

[0034] 图10是适于用来实现本申请实施例的电子设备的计算机系统的结构示意图。

## 具体实施方式

[0035] 这里将详细地对示例性实施例执行说明,其示例表示在附图中。下面的描述涉及

附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本申请相同的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本申请的一些方面相同的装置和方法的例子。

[0036] 本申请实施例中,术语“模块”或“单元”是指有预定功能的计算机程序或计算机程序的一部分,并与其他相关部分一起工作以实现预定目标,并且可以通过使用软件、硬件(如处理电路或存储器)或其组合来全部或部分实现。同样的,一个处理器(或多个处理器或存储器)可以用来实现一个或多个模块或单元。此外,每个模块或单元都可以是包含该模块或单元功能的整体模块或单元的一部分。

[0037] 附图中所示的方框图仅仅是功能实体,不一定必须与物理上独立的实体相对应。即,可以采用软件形式来实现这些功能实体,或在一个或多个硬件模块或集成电路中实现这些功能实体,或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

[0038] 附图中所示的流程图仅是示例性说明,不是必须包括所有的内容和操作/步骤,也不是必须按所描述的顺序执行。例如,有的操作/步骤还可以分解,而有的操作/步骤可以合并或部分合并,因此实际执行的顺序有可能根据实际情况改变。

[0039] 需要说明的是,在本申请中提及的“多个”是指两个或者两个以上。“和/或”描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。字符“/”一般表示前后关联对象是一种“或”的关系。

[0040] 随着互联网技术的发展,利用高效便捷性的业务系统处理业务已经是现代生活中常见的业务处理手段,业务系统通常可以整合多个业务流程,以满足业务目的的需求;例如企业管理系统,可以搭载有请假审批流程、货物管理流程,以及经费申请流程等。

[0041] 可以理解的是,业务流程的设计是业务系统搭建过程中的核心工作之一,其中业务流程引擎可以根据一定的协议将多个任务节点进行排列组合以创建/设计业务流程,业务流程引擎可以是BPMN标准流程引擎;例如Activiti/Flowable。Flowable是一种常用的业务流程引擎,其允许开发人员基于业务流程模型和标记法2.0(Business Process Model and Notation,BPMN)协议标准以流程图的形式设计业务流程,降低了业务流程创建/设计的难度。

[0042] 相关技术在基于业务流程引擎创建/设计好业务流程后,在业务流程的流转过程中,容易出现任务节点的任务驱动中断,造成业务流程执行失败的现象。为便于理解,请参阅图1,图1为业务流程执行失败的示意图。如图1所示,业务流程的执行过程中,任务节点B执行业务逻辑发生了异常中断,无法触发任务节点B任务的提交,从而导致无法流转到任务节点C,即此时业务流程流转失败。

[0043] 因此,为了提升业务流程的流转控制可靠性,本申请提供了一种业务流程的流转控制方案。请参阅图2,图2是本申请涉及的一种实施环境的示意图。该实施环境主要包括终端设备201和服务器202。

[0044] 其中,终端设备201可以预先安装用于管理企业的软件系统,该软件系统可以响应用户输入的操作执行某一具体的业务流程,以达到与该业务流程对应的业务目的,该软件系统可以依赖于业务流程引擎来实现对业务操作的管理。

[0045] 在业务流程执行过程中,该终端设备201可以将业务流程中的每个任务节点所调用的业务逻辑执行体(其用于执行业务逻辑)对应的存活信息发送给服务器202,以使服务

器202基于每个任务节点所调用的业务逻辑执行体对应的存活信息判断业务逻辑执行体在执行过程中是否出现异常中断,以便在判定业务逻辑执行体在执行过程中出现异常中断后,调用该异常中断的业务逻辑执行体重新执行业务逻辑,从而保证业务流程的正常执行。

[0046] 示例性地,终端设备201的类型包括但不限于智能手机、平板电脑、电视机、笔记本电脑、台式电脑等,对此不进行具体限定。

[0047] 其中,服务器202具有数据处理和存储功能。

[0048] 在业务流程执行过程中,服务器202可以接收终端设备201发送的业务流程中每个任务节点所调用的业务逻辑执行体对应的存活信息,并基于每个任务节点所调用的业务逻辑执行体对应的存活信息判断业务逻辑执行体在执行过程中是否出现异常中断,以便在判定业务逻辑执行体在执行过程中出现异常中断后,调用该异常中断的业务逻辑执行体重新执行业务逻辑,从而保证业务流程的正常执行。

[0049] 示例性地,服务器可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,其中服务器集群或者分布式系统包括用于提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、内容分发网络(Content Delivery Network,CDN)以及大数据和人工智能平台等基础云计算服务的云服务器,对此不进行具体限定。

[0050] 可以理解的是,终端设备201与服务器202之间通过有线或无线网络建立通信连接。示例性地,无线网络或有线网络使用标准通信技术和/或协议。网络通常为因特网、但也可以是其他任意网络,包括但不限于局域网(Local Area Network,LAN)、城域网(Metropolitan AreaNetwork,MAN)、广域网(Wide Area Network,WAN)、移动、有线或者无线网络、专用网络或者虚拟专用网络的任何组合等。

[0051] 应当明确的是,图2中的终端设备201和服务器202的数目仅仅是示意性的,根据实际需要,可以具有任意数量的终端设备201和服务器202。

[0052] 需要说明的是,在本申请的具体实施方式中,涉及到用户相关的数据,当本申请实施例运用到具体产品或技术中时,需要获得用户许可或者同意,且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准。

[0053] 以下对本申请实施例的技术方案的各种实现细节进行详细阐述:

[0054] 请参阅图3,图3是本申请的一个实施例示出的业务流程的流转控制方法的流程图,该业务流程的流转控制方法可以由服务器202来执行。如图3所示,该业务流程的流转控制方法至少包括S301至S304,详细介绍如下:

[0055] S301,若接收到针对业务流程引擎中预先创建的业务流程的启动请求,则启动业务流程,业务流程包括至少两个任务节点。

[0056] 可以理解的是,当用户针对业务流程引擎中预先创建的业务流程有执行需求时,其可以下发启动请求,并通过终端设备发送给服务器;相应地,服务器接收到针对业务流程引擎中预先创建的业务流程的启动请求,触发业务流程的启动,进而以通过业务流程的执行达到与该业务流程对应的业务目的。

[0057] 本申请实施例中业务流程包括至少两个任务节点,每个任务节点都对应有用于执行业务逻辑的业务逻辑执行体(也可称为业务执行器);其中,在业务流程的执行过程中,当流转到相应的任务节点时,任务节点可以调用业务逻辑执行体执行业务逻辑。

[0058] 举例说明,例如请再参阅图1,业务流程包括至少4个任务节点,分别为任务节点A、任务节点B、任务节点C,以及任务节点D,当流转到任务节点A时,其可以调用业务逻辑执行体a执行业务逻辑,当流转到任务节点B时,其可以调用业务逻辑执行体b执行业务逻辑,以此类推即可。

[0059] S302,获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,业务逻辑执行体用于执行业务逻辑。

[0060] 本申请实施例中在启动业务流程后,可以对业务流程当前所流转到的任务节点进行检测,具体地,可以是获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,并基于存活信息检测业务逻辑执行体的存活情况。

[0061] 本申请实施例中业务流程中当前所流转到的任务节点是指在业务流程执行过程中当前时间点所流转到的任务节点,其区别于流转过的任务节点和还未流转到的任务节点。

[0062] 举例说明,例如仍承接前述图1示例,设当前流转到任务节点B,那么任务节点A称为流转过的任务节点,任务节点C和任务节点D称为还未流转到的任务节点。

[0063] 本申请实施例中存活信息是指用于体现业务逻辑执行体是否存活的信息,其中如果存活信息表征业务逻辑执行体未存活,则意味着业务逻辑执行体被异常中断,即业务逻辑执行体执行业务逻辑被异常中断,而如果存活信息表征业务逻辑执行体存活,则意味着业务逻辑执行体在正常执行业务逻辑的过程中。

[0064] 本申请实施例中是获取当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息。

[0065] 举例说明,例如仍承接前述图1示例,如果流转到任务节点A,那么是获取任务节点A所调用的业务逻辑执行体a对应的存活信息,随着时间推移,如果流转到任务节点B,那么是获取任务节点B所调用的业务逻辑执行体b对应的存活信息,以此类推即可。

[0066] 在本申请的一个实施例中,S302中获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息的过程,可以包括:

[0067] 启动业务逻辑执行体检测服务;

[0068] 通过所启动的业务逻辑执行体检测服务周期性获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,并基于所获取到的存活信息检测业务逻辑执行体的存活情况。

[0069] 也即,可选实施例中服务器预置有用于获取存活信息并检测业务逻辑执行体存活情况的服务(称为业务逻辑执行体检测服务,也可称为中断检查服务),其中该业务逻辑执行体检测服务由开发人员或自动化机器预先编写得到。

[0070] 其中,可选实施例中服务器可以在启动业务流程的同时/之后/之前,启动业务逻辑执行体检测服务,进而在业务流程的执行过程中,由所启动的业务逻辑执行体检测服务周期性地获取当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,并基于所获取到的存活信息检测业务逻辑执行体的存活情况。

[0071] 换言之,业务逻辑执行体存活情况的检测是周期性的,这样通过持续检测能够及时发现出现异常中断的业务逻辑执行体,从而提升了对出现异常中断的业务逻辑执行体的再次调用效率,业务的执行效率高。

[0072] 这样通过实施可选实施例,由所启动的业务逻辑执行体检测服务能够简便准确地实现业务逻辑执行体存活情况的检测,为后续对出现异常中断的业务逻辑执行体的再次调用提供了有力支持。

[0073] S303,若基于存活信息检测到业务逻辑执行体异常中断,则再次调用业务逻辑执行体,以通过再次调用的业务逻辑执行体重新执行业务逻辑。

[0074] 本申请实施例中如果基于存活信息检测到业务逻辑执行体异常中断,则此时需要再次调用业务逻辑执行体,从而以通过再次调用的业务逻辑执行体重新执行业务逻辑。

[0075] 可以理解的是,如果基于存活信息检测到业务逻辑执行体正常执行,则此时无需做相应处理,按照正常业务流程执行即可。

[0076] 举例说明,例如仍承接前述图1示例,在业务流程的执行过程中:

[0077] 当流转到任务节点A时,是基于所获取到的任务节点A所调用的业务逻辑执行体a对应的存活信息对业务逻辑执行体a的存活情况进行检测,如果检测到任务节点A所调用的业务逻辑执行体a在正常执行业务逻辑,此时无需做相应处理,只需等待业务逻辑执行a执行完业务逻辑,以流转到任务节点B。

[0078] 当流转到任务节点B时,是基于所获取到的任务节点B所调用的业务逻辑执行体b对应的存活信息对业务逻辑执行体b的存活情况进行检测,如果检测到任务节点B所调用的业务逻辑执行体b异常中断,此时再次调用业务逻辑执行体b,从而以通过再次调用的业务逻辑执行体b重新执行业务逻辑。

[0079] 其他任务节点的处理过程以此类推即可。

[0080] 在本申请的一个实施例中,S303中再次调用业务逻辑执行体的过程,可以包括:

[0081] 调用业务流程引擎的业务重试接口;

[0082] 通过业务重试接口重新启动所流转到的任务节点,以基于所重新启动的任务节点再次调用业务逻辑执行体。

[0083] 也即,可选实施例中服务器是通过预先配置的业务流程引擎的业务重试接口对应的地址,调用业务重试接口,进而由所调用的业务重试接口重新启动所流转到的任务节点,以基于所重新启动的任务节点再次调用业务逻辑执行体执行业务逻辑。

[0084] 换言之,任务节点的重新启动带动任务节点所调用的业务逻辑执行体的重新启动,进而可以基于重新启动的业务逻辑执行体重新执行业务逻辑。

[0085] 这样通过实施可选实施例,由所调用的业务重试接口能够简便地实现业务逻辑执行体的再次调用,保证了业务流程的正常流转。

[0086] S304,在检测到执行完业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,以继续业务流程的流转。

[0087] 本申请实施例中如果执行完业务逻辑,则可以启动当前流转到的任务节点相邻的下一任务节点,从而以执行业务流程,保证业务流程的正常流转。

[0088] 举例说明,例如仍承接前述图1示例,在任务节点A调用业务逻辑执行体a执行完业务逻辑后,启动任务节点B,以使任务节点B调用业务逻辑执行体b执行业务逻辑(检测到业务逻辑执行体b发生异常中断后重新调用,直至业务逻辑执行b执行完业务逻辑);在任务节点B调用业务逻辑执行体a执行完业务逻辑后,启动任务节点C,以使任务节点B调用业务逻辑执行体b执行业务逻辑,以此类推即可。

[0089] 本申请实施例中通过业务逻辑执行体的存活信息实时检测业务逻辑执行体是否发生异常中断,并在其异常中断后对其进行再次调用,避免了业务逻辑执行体异常中断造成业务流程执行失败的现象,保证了业务流程的正常执行/流转,提升了业务流程的流转控制准确性、灵活性,业务流程的流转控制可靠性高。

[0090] 在本申请的一个实施例中,提供了另一种业务流程的流转控制方法,该业务流程的流转控制方法可以由服务器202来执行。如图4所示,该业务流程的流转控制方法可以包括S401至S402、S301、S303至S304。

[0091] S401至S402详细介绍如下:

[0092] S401,获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的标识信息。

[0093] 本申请实施例中在启动业务流程后,可以获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的标识信息。

[0094] 本申请实施例中业务逻辑执行体对应的标识信息用于唯一标识业务逻辑执行体,其包括但不限于业务逻辑执行体的名称、业务逻辑执行体的标识号等。

[0095] S402,从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息;其中,指定存储区域中存储的业务逻辑执行体的存活信息是业务逻辑执行体在执行业务逻辑过程中持续生成并发送的。

[0096] 本申请实施例中指定存储区域是指用于存储业务逻辑执行体的存活信息的区域。可以理解的是,业务逻辑执行体在执行业务逻辑过程中会持续生成业务逻辑执行体的存活信息,并将所生成的存活信息发送给服务器;相应地,服务器接收到业务逻辑执行体的存活信息,并将其存储到指定存储区域中。

[0097] 举例说明,例如仍承接前述图1示例,在业务流程的执行过程中:

[0098] 如果流转到任务节点A,则是获取任务节点A所调用的业务逻辑执行体a的标识信息“a”,并从指定存储区域中获取与业务逻辑执行体a的标识信息“a”相匹配的存活信息;其中,指定存储区域中存储的业务逻辑执行体a的存活信息是业务逻辑执行体a在执行业务逻辑过程中持续生成并发送给服务器的。

[0099] 如果流转到任务节点B,则是获取任务节点B所调用的业务逻辑执行体b的标识信息“b”,并从指定存储区域中获取与业务逻辑执行体b的标识信息“b”相匹配的存活信息;其中,指定存储区域中存储的业务逻辑执行体b的存活信息是业务逻辑执行体b在执行业务逻辑过程中持续生成并发送给服务器的。

[0100] 其他任务节点的处理过程以此类推即可。

[0101] 在本申请的一个实施例中,存活信息包括每个心跳消息发送的时间戳信息,心跳消息包括但不限于业务逻辑执行体的标识信息、业务逻辑执行体所属的业务流程的标识信息等。

[0102] 也即,可选实施例中存活信息包括的是心跳信息发送的时间戳信息,可选地,心跳信息的数量为多个,且每个心跳信息包括业务逻辑执行体的标识信息和业务逻辑执行体所属的业务流程的标识信息。

[0103] 举例说明,请参见表1,为一种示例的存活信息。

[0104] 表1

[0105]

| 业务逻辑执行体的标识信息 | 业务逻辑执行体所属的业务流程的标识信息 | 时间戳信息 |
|---|---|---|
| xxxx1 | kkkk1 | t1 |
| xxxx2 | kkkk2 | t2 |
| …… | …… | …… |

[0106] 相应地,S402中从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息的过程,可以包括:

[0107] 从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合;其中,同一存活信息集合包括同一业务逻辑执行体所针对的同一业务流程对应的多个时间戳信息。

[0108] 可以理解的是,由于实际应用中通常涉及到多个业务流程的执行,因此,服务器接收到的是来自于多个任务节点所调用的业务逻辑执行体所发送的业务逻辑执行体的存活信息。

[0109] 举例说明,例如针对业务流程1而言,设当前流转到任务节点$A_1$,则任务节点$A_1$所调用的业务逻辑执行体$a_1$在执行业务逻辑过程中会持续生成业务逻辑执行体$a_1$的存活信息,并将所生成的存活信息发送给服务器;相应地,服务器接收到业务逻辑执行体$a_1$的存活信息,并将其存储到指定存储区域中。

[0110] 同时针对业务流程2而言,设当前流转到任务节点$A_2$,则任务节点$A_2$所调用的业务逻辑执行体$a_2$在执行业务逻辑过程中会持续生成业务逻辑执行体$a_2$的存活信息,并将所生成的存活信息发送给服务器;相应地,服务器接收到业务逻辑执行体$a_2$的存活信息,并将其存储到指定存储区域中。

[0111] 因此,为了更好地管理业务逻辑执行体所发送的存活信息,在从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合的过程之前,还可以包括:

[0112] 基于存活信息中的业务逻辑执行体的标识信息和业务逻辑执行体所属的业务流程的标识信息,对存活信息进行分类,得到存活信息集合;其中,同一存活信息集合包括同一业务逻辑执行体所针对的同一业务流程对应的多个时间戳信息。

[0113] 也即,可选实施例中服务器是将相同业务逻辑执行体和相同业务流程对应的时间戳信息划分到同一个存活信息集合中,当然不同存活信息集合对应的业务逻辑执行体和业务流程均不相同。

[0114] 举例说明,例如承接前述示例,设分类得到存活信息集合N1和存活信息集合N2,其中存活信息集合N1包括业务流程1对应的业务逻辑执行体$a_1$的存活信息,存活信息集合N2包括业务流程2对应的业务逻辑执行体$a_2$的存活信息。

[0115] 可以理解的是,由于可选实施例中服务器预先对业务逻辑执行体的存活信息进行分类得到了多个存活信息集合;因此,当需要获取当前所流转到的任务节点的业务逻辑执行体对应的存活信息时,从指定存储区域所存储的多个存活信息集合中,选择与当前所流

转到的任务节点的业务逻辑执行体对应的标识信息,以及该业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合即可。

[0116] 这样通过实施可选实施例,预先对业务逻辑执行体的存活信息进行分类,不仅便于业务逻辑执行体的存活信息的管理,而且能够提升业务逻辑执行体的存活信息的获取效率,从而提升业务逻辑执行体是否异常中断的检测效率,业务的执行效率高。

[0117] 在本申请的一个实施例中,在从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合的过程之后,还可以包括:

[0118] 计算当前时间点与所选择出的存活信息集合中多个时间戳信息之间的间隔时长;

[0119] 基于所计算得到的间隔时长与预设间隔时长阈值之间的关系,检测业务逻辑执行体的存活情况。

[0120] 如前述实施例介绍,服务器会基于存活信息检测业务逻辑执行体的存活情况。具体地,可选实施例中服务器是基于当前时间点和业务逻辑执行体(即当前所流转到的任务节点对应的业务逻辑执行体)对应的存活信息集合中多个时间戳信息之间的间隔时长,实现业务逻辑执行体存活情况的检测。

[0121] 这样通过实施可选实施例,基于当前时间点和业务逻辑执行体对应的存活信息集合中多个时间戳信息之间的间隔时长,能够简便准确地实现业务逻辑执行体存活情况的检测,为后续对出现异常中断的业务逻辑执行体的再次调用提供了有力支持。

[0122] 在本申请的一个实施例中,计算当前时间点与所选择出的存活信息集合中多个时间戳信息之间的间隔时长的过程,可以包括:

[0123] 从所选择出的存活信息集合的多个时间戳信息中选择距离当前时间点最近的时间戳信息,并计算所选择出的时间戳信息与当前时间点之间的间隔时长;或者,

[0124] 计算当前时间点与所选择出的存活信息集合的每个时间戳信息之间的间隔时长,得到多个间隔时长,并从多个间隔时长中选择间隔时长最小的间隔时长。

[0125] 其中,可选实施例中服务器可以从所选择出的存活信息集合的多个时间戳信息中先选择出距离当前时间点最近的时间戳信息,之后再计算所选择出的时间戳信息与当前时间点之间的间隔时长。

[0126] 举例说明,例如设所选择出的存活信息集合N1={t1,t2,t3},其中t1,t2,t3均表征时间戳信息,t1>t2>t3,这里>表征早于,那么是从存活信息集合N1中选择出时间戳信息t3,并计算当前时间点t'与时间戳信息t3之间的间隔时长$\triangle t_3$。

[0127] 这样通过实施可选实施例,仅计算距离当前时间点最近的时间戳信息与当前时间点之间的间隔时长,节省了计算资源。

[0128] 其中,可选实施例中服务器可以先计算当前时间点与所选择出的存活信息集合的每个时间戳信息之间的间隔时长,得到多个间隔时长,之后再从多个间隔时长中选择间隔时长最小的间隔时长。

[0129] 举例说明,例如设所选择出的存活信息集合N1={t1,t2,t3},其中t1,t2,t3均表征时间戳信息,t1>t2>t3,这里>表征早于,此时可以计算当前时间点t'与时间戳信息t1之间的间隔时长$\triangle t_1$,计算当前时间点t'与时间戳信息t2之间的间隔时长$\triangle t_2$,以及计算当前时间点t'与时间戳信息t3之间的间隔时长$\triangle t_3$,之后再从3个间隔时长$\triangle t_1$,$\triangle t_2$,$\triangle t_3$中选择

出间隔时长最小的间隔时长(即$\triangle t_3$)。

[0130] 这样通过实施可选实施例,分别计算当前时间点与每个时间戳信息之间的间隔时长,流程简单,易于实施。

[0131] 在本申请的一个实施例中,基于所计算得到的间隔时长与预设间隔时长阈值之间的关系,检测业务逻辑执行体的存活情况的过程,可以包括:

[0132] 若所计算得到的间隔时长大于预设间隔时长阈值,则得到用于表征业务逻辑执行体异常中断的检测结果;

[0133] 若所计算得到的间隔时长等于或小于预设间隔时长阈值,则得到用于表征业务逻辑执行体正常执行的检测结果。

[0134] 也即,可选实施例中针对业务逻辑执行体存活情况的检测,存在两种情况:

[0135] 情况1,如果间隔时长大于预设间隔时长阈值,则表征业务逻辑执行体可能因异常中断而停止上报存活信息,此时确定业务逻辑执行体异常中断,即得到的是用于业务逻辑执行体异常中断的检测结果。

[0136] 情况2,如果间隔时长等于或小于预设间隔时长阈值,则表征业务逻辑执行体正常执行并上报存活信息,此时确定业务逻辑执行体正常执行,即得到的是用于业务逻辑执行体正常执行的检测结果。

[0137] 举例说明,例如承接前述示例,设预设间隔时长阈值为t0,如果$\triangle t_3 > t0$,则确定业务逻辑执行体异常中断,如果$\triangle t_3 \leqslant t0$,则确定业务逻辑执行体正常执行。

[0138] 这样通过实施可选实施例,将最小的间隔时长与预设间隔时长阈值进行比较,能够简便准确地实现业务逻辑执行体存活情况的检测。

[0139] 在本申请的一个实施例中,还可以包括:

[0140] 接收业务逻辑执行体发送的执行完成通知信息;其中,执行完成通知信息是在业务逻辑执行体执行完业务逻辑后生成的;

[0141] 基于执行完成通知信息从指定存储区域中删除业务逻辑执行体的存活信息。

[0142] 也即,可选实施例中业务逻辑执行体在执行完业务逻辑后可以生成执行完成通知信息,其中执行完成通知信息用于表征业务逻辑执行体执行完业务逻辑,并将执行完成通知信息发送给服务器;相应地,服务器接收到业务逻辑执行体发送的执行完成通知信息,能够明确业务逻辑执行体已执行完业务逻辑,从而可以从指定存储区域中将业务逻辑执行体的存活信息予以删除。

[0143] 这样通过实施可选实施例,业务逻辑执行体在执行完业务逻辑后及时通知服务器,服务器将指定存储区域中业务逻辑执行体的存活信息进行删除,节省了存储资源,也在一定程度上提升了后续业务逻辑执行体的存活信息的获取效率/查找效率。

[0144] 在其他实施例中,还可以是服务器在检测到业务逻辑执行体执行完业务逻辑后,主动触发指定存储区域中业务逻辑执行体存活信息的删除,在实际应用中,可以根据具体应用场景进行灵活调整。

[0145] 需要说明的是,图4所示中S301、S303至S304的详细介绍请参见图3所示的S301、S303至S304,在此不再赘述。

[0146] 本申请实施例中通过业务逻辑执行体对应的标识信息能够快速地从指定存储区域中获取到业务逻辑执行体的存活信息,提升了业务逻辑执行体的存活信息的获取效率,

从而提升了业务逻辑执行体是否异常中断的检测效率,业务的执行效率高。

[0147] 在本申请的一个实施例中,提供了另一种业务流程的流转控制方法,该业务流程的流转控制方法可以由服务器202来执行。如图5所示,该业务流程的流转控制方法在S301之后还可以包括S501至S503。

[0148] 应当明确的是,前述实施例中介绍的业务逻辑执行体出现异常中断,通常是指硬中断,其具有强制中断的特性),其与软中断为相对概念,软中断具有优雅退出的特性,即可以等待一段时长后再退出。本申请实施例中将介绍针对软中断的处理过程。

[0149] S501至S503详细介绍如下:

[0150] S501,若接收到软中断请求,则获取当前处于执行状态的业务逻辑执行体。

[0151] 可以理解的是,由于实际应用中通常涉及到多个业务流程的执行,因此,为了保证每个业务流程的正常流转,在接收到软中断请求后,可以获取当前处于执行状态的业务逻辑执行体,以基于当前处于执行状态的业务逻辑执行体执行相应的等待操作。

[0152] 本申请实施例中处于执行状态的业务逻辑执行体是指正在执行业务逻辑的业务逻辑执行体。

[0153] 在本申请的一个实施例中,S501中获取当前处于执行状态的业务逻辑执行体的过程,可以包括:

[0154] 获取用于记录处于执行状态的业务逻辑执行体的数量的数量参数;其中,数量参数是通过在每调用一个业务逻辑执行体时进行指定单位量的增加,以及在每一个业务逻辑执行体执行完业务逻辑时进行指定单位量的减少所得到的;

[0155] 若检测到数量参数所表征的处于执行状态的业务逻辑执行体的数量为至少一个,则确定当前存在处于执行状态的业务逻辑执行体。

[0156] 也即,可选实施例中服务器预先设置有一个用于记录处于执行状态的业务逻辑执行体的数量的数量参数(例如counter),并会持续对业务流程过程中业务逻辑执行体的调用进行检测。

[0157] 其中,可选实施例中是每当检测到调用了一个业务逻辑执行体,则对处于执行状态的业务逻辑执行体的数量进行增加,通常为counter=couner+1,当然也可以增加其他指定单位量,例如counter=couner+2等;而当检测到每一个业务逻辑执行体执行完了业务逻辑,则对处于执行状态的业务逻辑执行体的数量进行减少,通常为counter=couner-1,当然也可以减少其他指定单位量,例如counter=couner-2。可以理解的是,增加和减少所针对的单位量相同。

[0158] 其中,可选实施例中如果检测到数量参数所表征的处于执行状态的业务逻辑执行体的数量为至少一个,则此时存在正在执行业务逻辑的业务逻辑执行体,而如果检测到数量参数所表征的处于执行状态的业务逻辑执行体的数量为零,则此时不存在正在执行业务逻辑的业务逻辑执行体。

[0159] 这样通过实施可选实施例,通过数量参数的设置能够明确当前处于执行状态的业务逻辑执行体的数量,为业务流程的退出控制提供了有力支持。

[0160] S502,等待处于执行状态的业务逻辑执行体执行完业务逻辑。

[0161] 本申请实施例中当存在正在执行业务逻辑的业务逻辑执行体,则可以等待该业务逻辑执行体执行完业务逻辑。

[0162] S503,在处于执行状态的业务逻辑执行体执行完业务逻辑后,控制退出业务流程。

[0163] 本申请实施例中服务器可以持续对业务逻辑执行体针对业务逻辑的执行情况进行检测,如果检测到处于执行状态的业务逻辑执行体执行完业务逻辑后,则可以控制退出业务流程,即无需再等待,而如果检测到处于执行状态的业务逻辑执行体仍在执行业务逻辑,则可以继续等待。

[0164] 在本申请的一个实施例中,S503中在处于执行状态的业务逻辑执行体执行完业务逻辑后,控制退出业务流程的过程,可以包括:

[0165] 若检测到参数所表征的处于执行状态的业务逻辑执行体的数量为零,则控制退出业务流程。

[0166] 也即,可选实施例中如果检测到数量参数所表征的处于执行状态的业务逻辑执行体的数量为零,则此时不存在正在执行业务逻辑的业务逻辑执行体,控制退出业务流程即可。

[0167] 在本申请的一个实施例中,在S502中等待处于执行状态的业务逻辑执行体执行完业务逻辑的过程,可以包括:

[0168] 若检测到触发等待处于执行状态的业务逻辑执行体执行完业务逻辑,则记录等待的开始时间点,以及从开始时间点进行计时;

[0169] 相应地,还可以包括:若检测到计时时长达到预设计时时长阈值,则控制退出业务流程。

[0170] 也即,可选实施例中服务器是在触发等待处于执行状态的业务逻辑执行体执行完业务逻辑时,记录等待的开始时间点,并从开始时间点进行计时,进而当检测到计时时长达到预设计时时长阈值,则控制退出业务流程。

[0171] 换言之,针对软中断设置有超时时长(即预设计时时长阈值),当软中断等待超过了超时时长,也会对其进行强制中断,即此时属于硬中断。

[0172] 这样通过实施可选实施例,通过超时时长的设置,以及利用计时时长与超时时长的比较控制业务流程的退出,避免了漫长等待的现象,平衡了业务执行和退出管理,适用于诸多场景中。

[0173] 需要说明的是,图5所示中S301至S304的详细介绍请参见图3所示的S301至S304,在此不再赘述。

[0174] 本申请实施例中在软中断的情况下,等待处于执行状态的业务逻辑执行体执行完业务逻辑后,才控制退出业务流程,保证了业务流程的正常流转。

[0175] 以下对本申请实施例的具体场景进行详细说明:

[0176] DevOps网络运营流程平台(NetDevOps):一种自动化运营流程开发体系与运行平台,具体地,是腾讯网络平台部为网络运营人员量身打造的自动化运营流程开发体系与运行平台;其具有通过低代码、低门槛、高效率的流程开发方式,流程化与标准化网络运营事务,以及统一的模式提供数据管理、自动化操作、以及事务管控等能力。

[0177] 自研工作流引擎(WorkflowEngine):一种为NetDevOps量身打造的自研工作流引擎,主要负责业务流程在运行过程中对任务的流转驱动,以及对流经任务节点的任务逻辑分发;其具有包括但不限于流程实例/任务/顺序流/变量的存储与管理、流程路径决策、任务逻辑分发、各类业务执行器执行、流程异常控制等能力。

[0178] 本申请实施例中以自研工作流引擎中所创建/设计的业务流程为例。

[0179] 请参阅图6,自研工作流引擎主要包括自研工作流引擎服务进程和中断检测服务进程,其中:

[0180] 系统初始化阶段(即无业务流程启动):自研工作流引擎服务进程设置全局计数器counter(即数量参数),并初始化为0,即表征当前没有启动的业务执行器对应的工作进程worker。

[0181] 系统运行阶段(即存在业务流程启动):所启动的业务流程对应的任务节点异步启动业务执行单元对应的工作进程worker。首先,针对业务执行单元对应的工作进程worker的初始阶段,对全局计数器counter+1,即表征当前增加了一个已启动的业务执行器;之后,针对业务执行单元对应的工作进程worker的执行阶段,调用业务执行器执行业务逻辑,并返回业务逻辑的执行结果;之后,针对业务执行单元对应的工作进程worker的退出阶段,对全局计数器counter-1,即表征当前减少了一个已启动的业务执行器。

[0182] 其中,针对业务执行单元对应的工作进程worker的初始阶段,还会异步启动心跳单元,进而业务执行单元对应的工作进程worker持续向心跳单元上报心跳,心跳单元接收到业务执行单元对应的工作进程worker上报的心跳,将心跳存储到中央标记存储中;中断检测服务进程从中央标记存储中读取业务执行单元对应的工作进程worker所上报的心跳进行超时检测/计算,并在确定超时时重新启动业务执行单元对应的工作进程worker(即带动业务执行器的重新启动)。

[0183] 可以理解的是,如果超时则意味着硬中断,自研工作流引擎服务进程强制退出,处于同一个服务进程不同线程或协程对应的心跳单元同样会被强制退出,即心跳停止,因而,在硬中断的情况下重新启动任务节点,以重新启动业务执行单元对应的工作进程worker(即带动业务执行器的重新启动)。

[0184] 其中,针对业务执行单元对应的工作进程worker的退出阶段,会结束向心跳单元上报心跳(例如业务执行单元对应的工作进程worker向心跳单元发送心跳取消信息,以取消心跳上报),之后业务执行单元对应的工作进程worker和心跳单元退出。可选地,心跳单元(退出前)和/或中断检测服务进程可以将中央标记存储中该业务执行单元对应的工作进程worker所上报的心跳进行删除。

[0185] 系统退出阶段(即接收到软中断请求例如singal.SIGTERM后):自研工作流引擎服务进程等待全局计数器counter变为0后退出,即表征当前没有启动的业务执行单元对应的工作进程worker,此时安全退出。可选地,在等待过程中如果检测到超时则强制退出(即硬中断)。

[0186] 需要说明的是,硬中断和软中断的区别在于硬中断无法感知到,软中断能够感知到,因此,本申请实施例中针对硬中断采用的是心跳检测方案,而针对软中断采用的是全局计数检测方案。

[0187] 本申请实施例中结合心跳检测方案和全局计数检测方案解决了业务执行器异步执行过程中的中断问题,保证了业务执行器的正常执行,从而保证了业务流程的正常流转,提升了业务流程的流转控制可靠性。

[0188] 接下来,详细介绍任务节点的相关流程。

[0189] 请参阅图7,图7是本申请的一个实施例示出的业务流程的流转控制方法的流程

图。如图7所示,该业务流程的流转控制方法至少包括S701至S711,详细介绍如下:

[0190] S701,任务节点异步启动异步协程C01。

[0191] S702,异步协程C01通过全局计数器进行计数(即counter+1)。

[0192] S703,异步协程C01启动业务执行器,通过业务执行器执行业务逻辑。

[0193] S704,异步协程C01检测到业务执行器执行完业务逻辑后,根据业务逻辑执行结果进行回调,以继续业务流程的流转。

[0194] S705,异步协程C01检测到回调成功,则确定异步协程C01已完成调用。

[0195] S706,异步协程C01更新全局计数器(即counter-1)。

[0196] S707,异步协程C01退出,并向异步协程C02发送心跳取消信息。

[0197] S708,异步协程C01异步启动异步协程C02。

[0198] S709,异步协程C02进行行为判断;如果接收到业务执行器的心跳,则执行S710,如果接收到心跳取消信息,则执行S711。

[0199] 可以理解的是,异步协程C02进行循环判断,直至接收到异步协程C01发送的心跳取消信息。

[0200] S710,异步协程C02将业务执行器的心跳存储到中央标记存储中。

[0201] S711,异步协程C02设置该业务执行器执行结束标记,并将该业务执行器的执行结束标记发送给中央标记存储,以使中央标记存储基于该业务执行器的执行结束标记删除该业务执行器的心跳,异步协程C02退出。

[0202] 需要说明的是,图7所示中S702至S707,以及S708至S711可以并行执行,同时S701至S711的详细介绍请参见前述实施例介绍,在此不再赘述。

[0203] 本申请实施例中任务节点通过两个异步协程共同实现心跳检测和全局计数检测,流程简单,应用场景广泛。

[0204] 接下来,详细介绍中断检测服务进程的相关流程。

[0205] 请参阅图8,图8是本申请的一个实施例示出的业务流程的流转控制方法的流程图。如图8所示,该业务流程的流转控制方法至少包括S801至S805,详细介绍如下:

[0206] S801,中断检测服务进程向中央标记存储发送用于查询业务执行器的存活信息的查询请求。

[0207] 可以理解的是,中断检测服务进程向中央标记存储发送用于查询业务执行器的存活信息的查询请求是持续性的,例如每隔3秒、5秒、10秒等发送一次,以实时检测业务执行器的存活情况,其中发送时长越短则检测的及时性越高,反之,发送时长越长则检测的及时性越低。

[0208] S802,中央标记存储基于查询请求向中断检测服务进程返回该业务执行器的存活信息。

[0209] 其中,该业务执行器的存活信息包括业务执行器每个心跳发送的时间戳信息,心跳包括该业务执行器的标识和该业务执行器所属的业务流程的标识。

[0210] 示例性地,该业务执行器的存活信息可以是业务执行器每隔5秒向中央标记存储上报的。其中存活信息的上报时间可以根据具体应用场景进行灵活调整。

[0211] S803,中断检测服务进程基于业务执行器的存活信息进行心跳时效性检测;如果检测到当前时间点与最近心跳的时间戳信息之间的间隔时长大于预设间隔时长,则执行

S804,如果检测到当前时间点与最近心跳的时间戳信息之间的间隔时长等于或小于预设间隔时长,则结束流程。

[0212] 示例性地,承接前述示例,如果业务执行器存活(即业务执行器正常执行业务逻辑),理论上,当前时间点与最近心跳的时间戳信息之间的间隔时长小于或等于5秒,而如果当前时间点与最近心跳的时间戳信息之间的间隔时长大于5秒,则意味着业务执行器未存活(即被异常中断,业务执行器未正常执行业务逻辑)。

[0213] 可选实施例中可以将预设间隔时长设置为20秒,以避免因网络出现超时所导致的误判现象,从而提升心跳时效性检测的准确性。

[0214] S804,基于该业务执行器所属的业务流程的标识调用工作流引擎的任务重试接口,启动任务节点,以基于所重新启动的任务节点再次调用业务执行器执行业务逻辑。

[0215] S805,工作流引擎返回任务重试接口调用结果。

[0216] 本申请实施例中中断检测服务进程在检测到业务执行器异常中断后,重新调用业务执行器执行业务逻辑,保证了业务流程的正常流转,提升了业务流程的流转控制可靠性。

[0217] 图9是本申请的一个实施例示出的业务流程的流转控制装置的框图。如图9所示,该业务流程的流转控制装置包括:

[0218] 第一启动模块901,配置为若接收到针对业务流程引擎中预先创建的业务流程的启动请求,则启动所述业务流程,所述业务流程包括至少两个任务节点;

[0219] 获取模块902,配置为获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,所述业务逻辑执行体用于执行业务逻辑;

[0220] 调用模块903,配置为若基于所述存活信息检测到所述业务逻辑执行体异常中断,则再次调用所述业务逻辑执行体,以通过再次调用的业务逻辑执行体重新执行所述业务逻辑;

[0221] 第二启动模块904,配置为在检测到执行完所述业务逻辑后,启动所流转到的任务节点相邻的下一任务节点,以继续所述业务流程的流转。

[0222] 在本申请的一个实施例中,基于前述方案,获取模块902,具体配置为:

[0223] 获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的标识信息;

[0224] 从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息;其中,所述指定存储区域中存储的所述业务逻辑执行体的存活信息是所述业务逻辑执行体在执行业务逻辑过程中持续生成并发送的。

[0225] 在本申请的一个实施例中,基于前述方案,所述存活信息包括每个心跳消息发送的时间戳信息,所述心跳消息包括所述业务逻辑执行体的标识信息和所述业务逻辑执行体所属的业务流程的标识信息;获取模块902,还具体配置为:

[0226] 从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合;其中,同一存活信息集合包括同一业务逻辑执行体所针对的同一业务流程对应的多个时间戳信息。

[0227] 在本申请的一个实施例中,基于前述方案,在所述从指定存储区域所存储的多个存活信息集合中,选择与所获取到的业务逻辑执行体对应的标识信息,以及所获取到的业

务逻辑执行体所属的业务流程对应的标识信息均相匹配的存活信息集合之后,所述装置还包括:

[0228] 计算模块,配置为计算当前时间点与所选择出的存活信息集合中多个时间戳信息之间的间隔时长;

[0229] 检测模块,配置为基于所计算得到的间隔时长与预设间隔时长阈值之间的关系,检测所述业务逻辑执行体的存活情况。

[0230] 在本申请的一个实施例中,基于前述方案,所述计算模块,具体配置为:

[0231] 从所选择出的存活信息集合的多个时间戳信息中选择距离当前时间点最近的时间戳信息,并计算所选择出的时间戳信息与所述当前时间点之间的间隔时长;或者

[0232] 计算所述当前时间点与所选择出的存活信息集合的每个时间戳信息之间的间隔时长,得到多个间隔时长,并从所述多个间隔时长中选择间隔时长最小的间隔时长。

[0233] 在本申请的一个实施例中,基于前述方案,所述检测模块,具体配置为:

[0234] 若所计算得到的间隔时长大于预设间隔时长阈值,则得到用于表征所述业务逻辑执行体异常中断的检测结果;

[0235] 若所计算得到的间隔时长等于或小于所述预设间隔时长阈值,则得到用于表征所述业务逻辑执行体正常执行的检测结果。

[0236] 在本申请的一个实施例中,基于前述方案,所述装置还包括:

[0237] 接收模块,配置为接收所述业务逻辑执行体发送的执行完成通知信息;其中,所述执行完成通知信息是在所述业务逻辑执行体执行完业务逻辑后生成的;

[0238] 删除模块,配置为基于所述执行完成通知信息从所述指定存储区域中删除所述业务逻辑执行体的存活信息。

[0239] 在本申请的一个实施例中,基于前述方案,获取模块902,具体配置为:

[0240] 启动业务逻辑执行体检测服务;

[0241] 通过所启动的业务逻辑执行体检测服务周期性获取所述业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息,并基于所获取到的存活信息检测所述业务逻辑执行体的存活情况。

[0242] 在本申请的一个实施例中,基于前述方案,调用模块903,具体配置为:

[0243] 调用所述业务流程引擎的业务重试接口;

[0244] 通过所述业务重试接口重新启动所流转到的任务节点,以基于所重新启动的任务节点再次调用所述业务逻辑执行体。

[0245] 在本申请的一个实施例中,基于前述方案,所述装置还包括退出控制模块,配置为:

[0246] 若接收到软中断请求,则获取当前处于执行状态的业务逻辑执行体;

[0247] 等待所述处于执行状态的业务逻辑执行体执行完业务逻辑;

[0248] 在所述处于执行状态的业务逻辑执行体执行完业务逻辑后,控制退出所述业务流程。

[0249] 在本申请的一个实施例中,基于前述方案,所述退出控制模块,具体配置为:

[0250] 获取用于记录处于执行状态的业务逻辑执行体的数量的数量参数;其中,所述数量参数是通过在每调用一个业务逻辑执行体时进行指定单位量的增加,以及在每一个业务

逻辑执行体执行完业务逻辑时进行所述指定单位量的减少所得到的;

[0251] 若检测到所述数量参数所表征的处于执行状态的业务逻辑执行体的数量为至少一个,则确定当前存在处于执行状态的业务逻辑执行体;

[0252] 若检测到所述数量参数所表征的处于执行状态的业务逻辑执行体的数量为零,则控制退出所述业务流程。

[0253] 在本申请的一个实施例中,基于前述方案,所述退出控制模块,还具体配置为:

[0254] 若检测到触发等待所述处于执行状态的业务逻辑执行体执行完业务逻辑,则记录等待的开始时间点,以及从所述开始时间点进行计时;

[0255] 若检测到计时时长达到预设计时时长阈值,则控制退出所述业务流程。

[0256] 需要说明的是,前述实施例所提供的装置与前述实施例所提供的方法属于同一构思,其中各个模块和单元执行操作的具体方式已经在方法实施例中进行了详细描述。

[0257] 本申请的实施例还提供了一种电子设备,包括:一个或多个处理器;存储器,用于存储一个或多个程序,当一个或多个程序被一个或多个处理器执行时,使得电子设备实现如前的业务流程的流转控制方法。

[0258] 图10是适于用来实现本申请实施例的电子设备的计算机系统的结构示意图。

[0259] 需要说明的是,图10示出的电子设备的计算机系统1000仅是一个示例,不应对本申请实施例的功能和使用范围带来任何限制。

[0260] 如图10所示,计算机系统1000包括中央处理单元(Central Processing Unit,CPU)1001,其可以根据存储在只读存储器(Read-Only Memory,ROM)1002中的程序或者从存储部分1008加载到随机访问存储器(Random Access Memory,RAM)1003中的程序而执行各种适当的动作和处理,例如执行上述实施例中的方法。在RAM 1003中,还存储有系统操作所需的各种程序和数据。CPU 1001、ROM 1002以及RAM 1003通过总线1004彼此相连。输入/输出(Input /Output,I/O)接口1005也连接至总线1004。

[0261] 以下部件连接至I/O接口1005:包括键盘、鼠标等的输入部分1006;包括诸如阴极射线管(Cathode Ray Tube,CRT)、液晶显示器(Liquid Crystal Display,LCD)等以及扬声器等的输出部分1007;包括硬盘等的存储部分1008;以及包括诸如LAN(Local Area Network,局域网)卡、调制解调器等的网络接口卡的通信部分1009。通信部分1009经由诸如因特网的网络执行通信处理。驱动器1010也根据需要连接至I/O接口1005。可拆卸介质1011,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器1010上,以便于从其上读出的计算机程序根据需要被安装入存储部分1008。

[0262] 特别地,根据本申请的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本申请的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的计算机程序。在这样的实施例中,该计算机程序可以通过通信部分1009从网络上被下载和安装,和/或从可拆卸介质1011被安装。在该计算机程序被中央处理单元(CPU)1001执行时,执行本申请的系统中限定的各种功能。

[0263] 需要说明的是,本申请实施例所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读介质例如可以是电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读介质的

更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(Erasable Programmable Read Only Memory,EPROM)、闪存、光纤、便携式紧凑磁盘只读存储器(Compact Disc Read-Only Memory,CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本申请中,计算机可读介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本申请中,计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的计算机程序。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的计算机程序可以用任何适当的介质传输,包括但不限于:无线、有线等等,或者上述的任意合适的组合。

[0264] 附图中的流程图和框图,图示了按照本申请各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。其中,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不相同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的是,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0265] 描述于本申请实施例中所涉及到的单元可以通过软件的方式实现,也可以通过硬件的方式来实现,所描述的单元也可以设置在处理器中。其中,这些单元的名称在某种情况下并不构成对该单元本身的限定。

[0266] 本申请的另一方面还提供了一种计算机可读介质,其上存储有计算机程序,该计算机程序被处理器执行时实现如前的业务流程的流转控制方法。该计算机可读介质可以是上述实施例中描述的电子设备中所包含的,也可以是单独存在,而未装配入该电子设备中。

[0267] 本申请的另一方面还提供了一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读介质中。计算机设备的处理器从计算机可读介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行上述各个实施例中提供的业务流程的流转控制方法。

[0268] 上述内容,仅为本申请的较佳示例性实施例,并非用于限制本申请的实施方案,本领域普通技术人员根据本申请的主要构思和精神,可以十分方便地进行相应的变通或修改,故本申请的保护范围应以权利要求书所要求的保护范围为准。

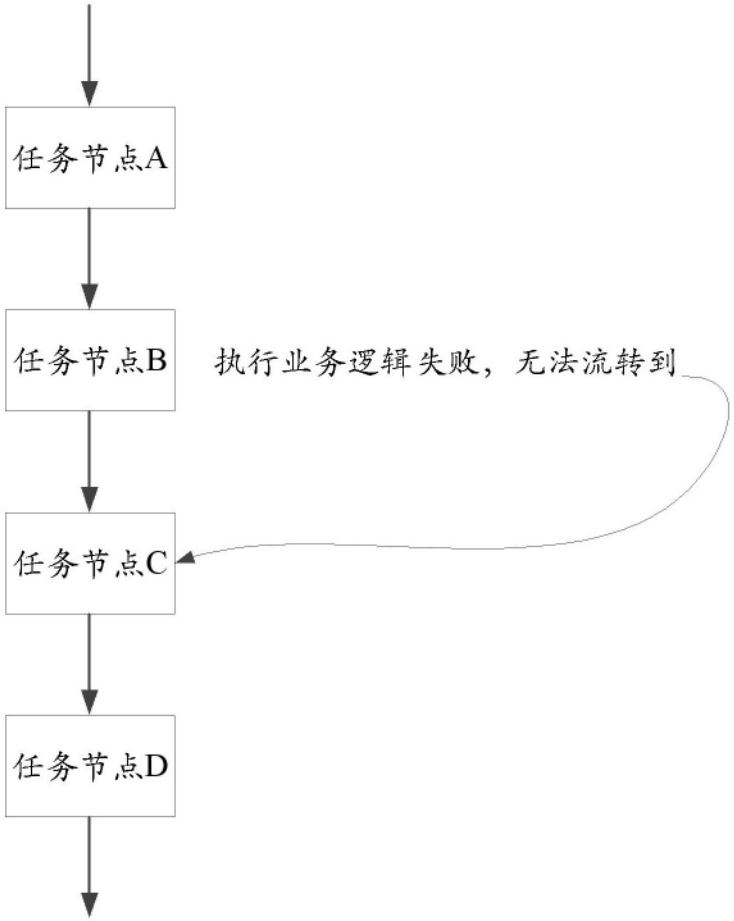



图1

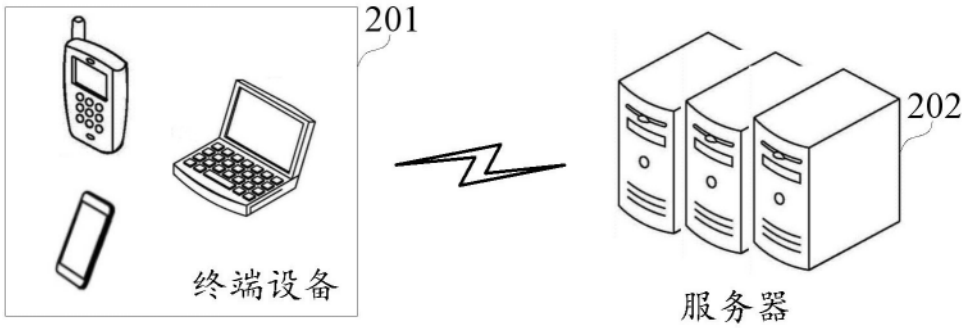



图2

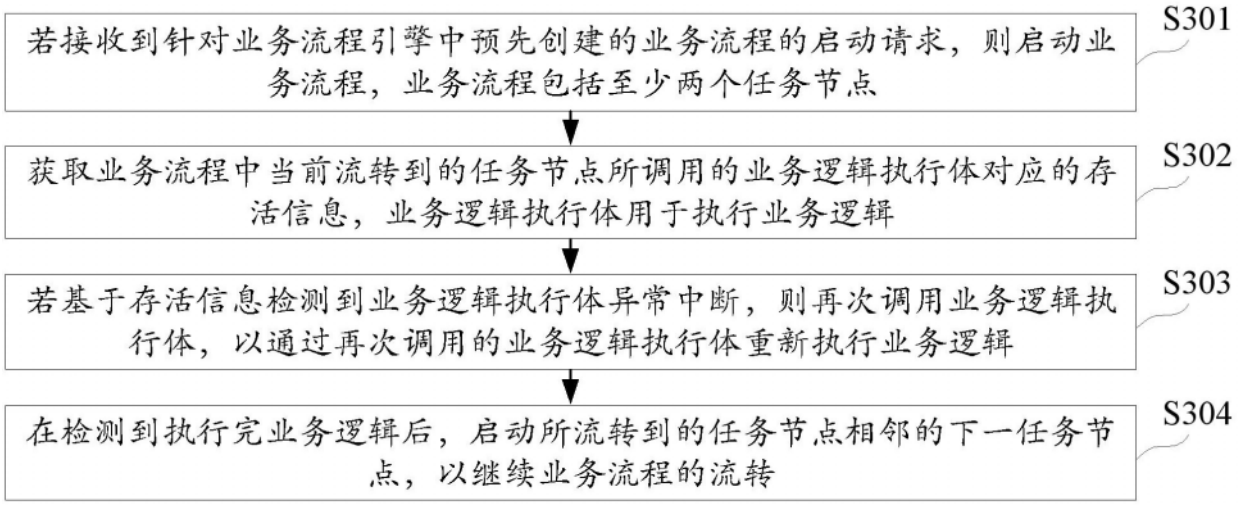

若接收到针对业务流程引擎中预先创建的业务流程的启动请求，则启动业务流程，业务流程包括至少两个任务节点
S301
获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息，业务逻辑执行体用于执行业务逻辑
S302
若基于存活信息检测到业务逻辑执行体异常中断，则再次调用业务逻辑执行体，以通过再次调用的业务逻辑执行体重新执行业务逻辑
S303
在检测到执行完业务逻辑后，启动所流转到的任务节点相邻的下一任务节点，以继续业务流程的流转
S304

图3

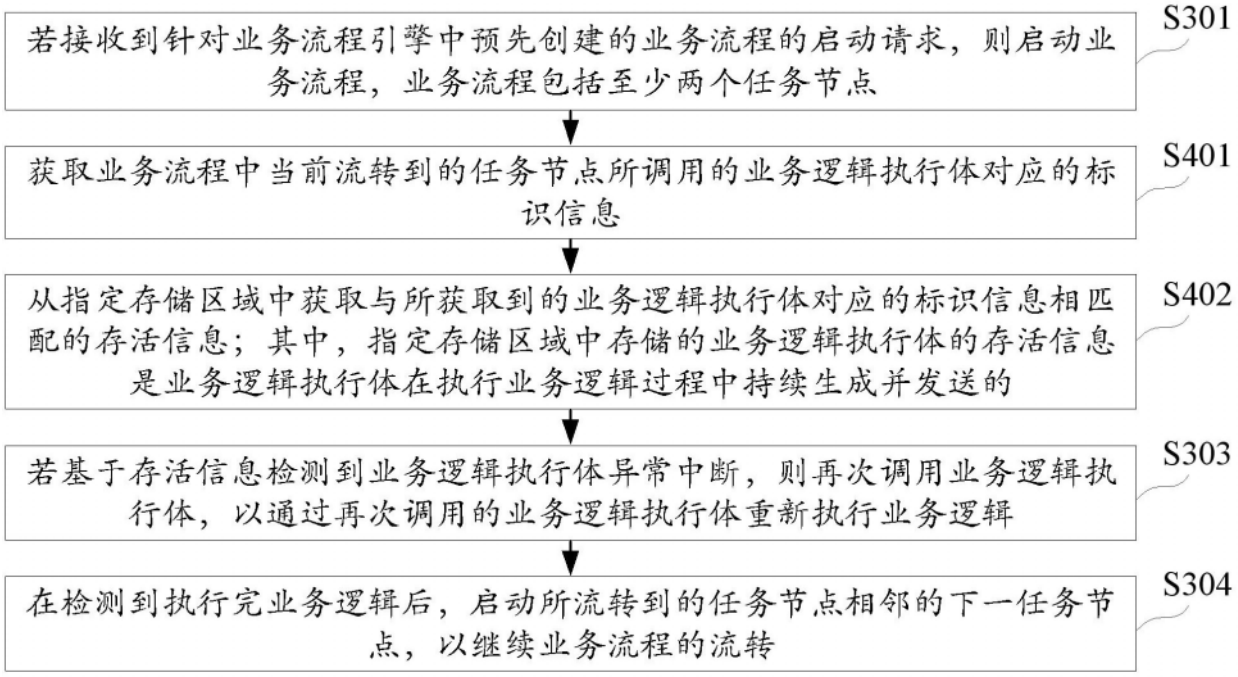

若接收到针对业务流程引擎中预先创建的业务流程的启动请求，则启动业务流程，业务流程包括至少两个任务节点
S301
获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的标识信息
S401
从指定存储区域中获取与所获取到的业务逻辑执行体对应的标识信息相匹配的存活信息；其中，指定存储区域中存储的业务逻辑执行体的存活信息是业务逻辑执行体在执行业务逻辑过程中持续生成并发送的
S402
若基于存活信息检测到业务逻辑执行体异常中断，则再次调用业务逻辑执行体，以通过再次调用的业务逻辑执行体重新执行业务逻辑
S303
在检测到执行完业务逻辑后，启动所流转到的任务节点相邻的下一任务节点，以继续业务流程的流转
S304

图4

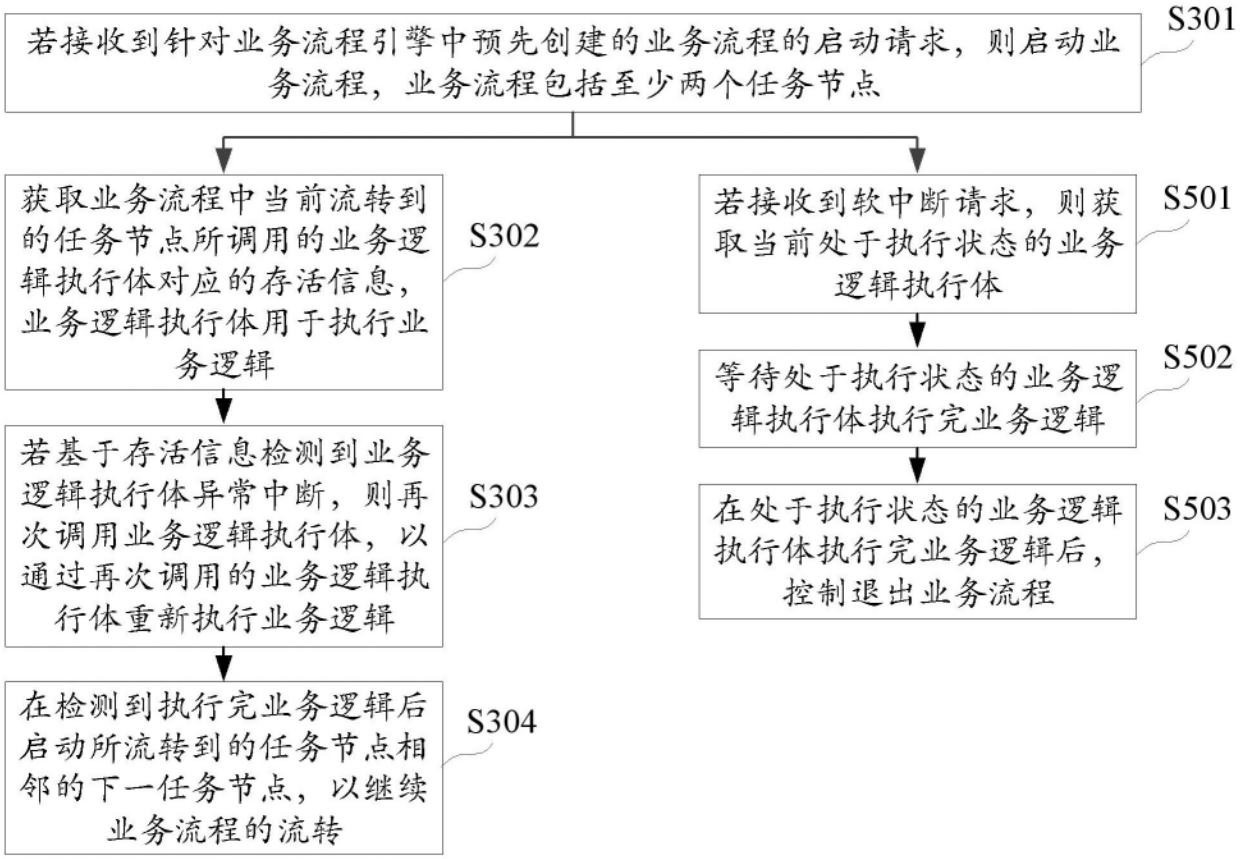

若接收到针对业务流程引擎中预先创建的业务流程的启动请求，则启动业务流程，业务流程包括至少两个任务节点
S301
获取业务流程中当前流转到的任务节点所调用的业务逻辑执行体对应的存活信息，业务逻辑执行体用于执行业务逻辑
S302
若基于存活信息检测到业务逻辑执行体异常中断，则再次调用业务逻辑执行体，以通过再次调用的业务逻辑执行体重新执行业务逻辑
S303
在检测到执行完业务逻辑后启动所流转到的任务节点相邻的下一任务节点，以继续业务流程的流转
S304
若接收到软中断请求，则获取当前处于执行状态的业务逻辑执行体
S501
等待处于执行状态的业务逻辑执行体执行完业务逻辑
S502
在处于执行状态的业务逻辑执行体执行完业务逻辑后，控制退出业务流程
S503

图5

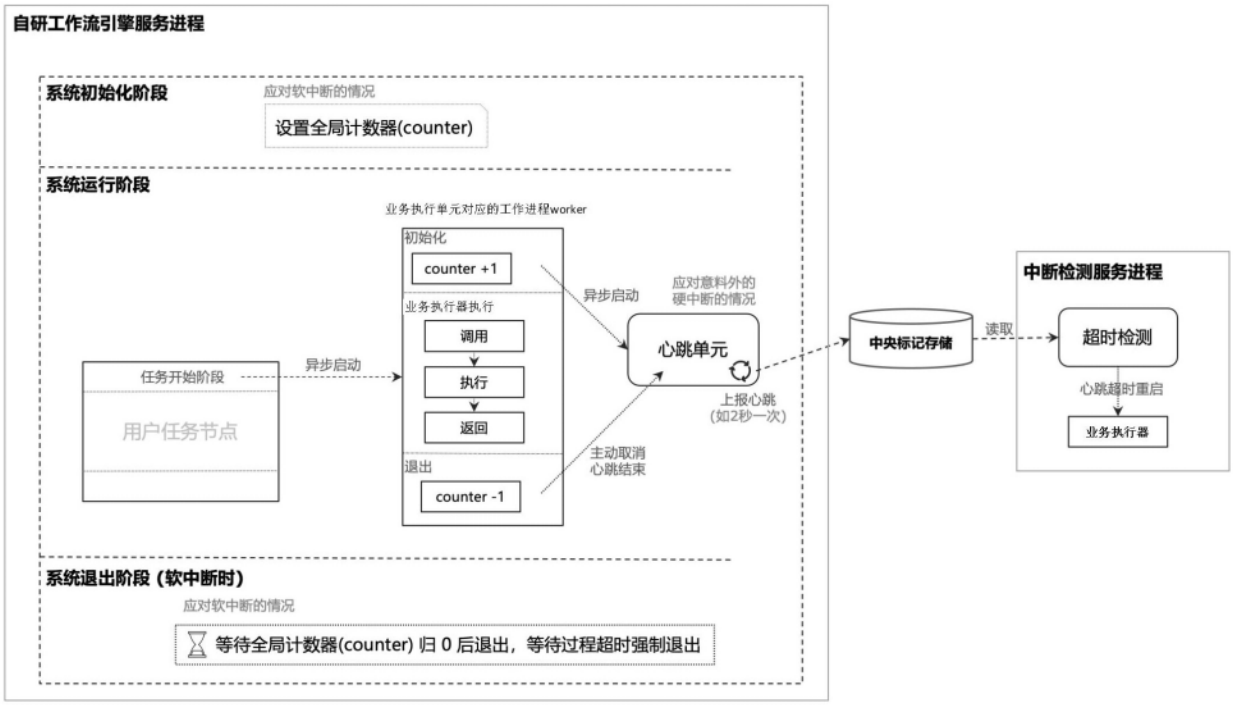

自研工作流引擎服务进程
系统初始化阶段
应对软中断的情况
设置全局计数器(counter)
系统运行阶段
业务执行单元对应的工作进程worker
初始化
counter +1
异步启动
应对意料外的硬中断的情况
业务执行器执行
调用
执行
返回
心跳单元
任务开始阶段
异步启动
用户任务节点
退出
counter -1
主动取消心跳结束
上报心跳(如2秒一次)
中央标记存储
读取
中断检测服务进程
超时检测
心跳超时重启
业务执行器
系统退出阶段（软中断时）
应对软中断的情况
等待全局计数器(counter) 归 0 后退出，等待过程超时强制退出

图6

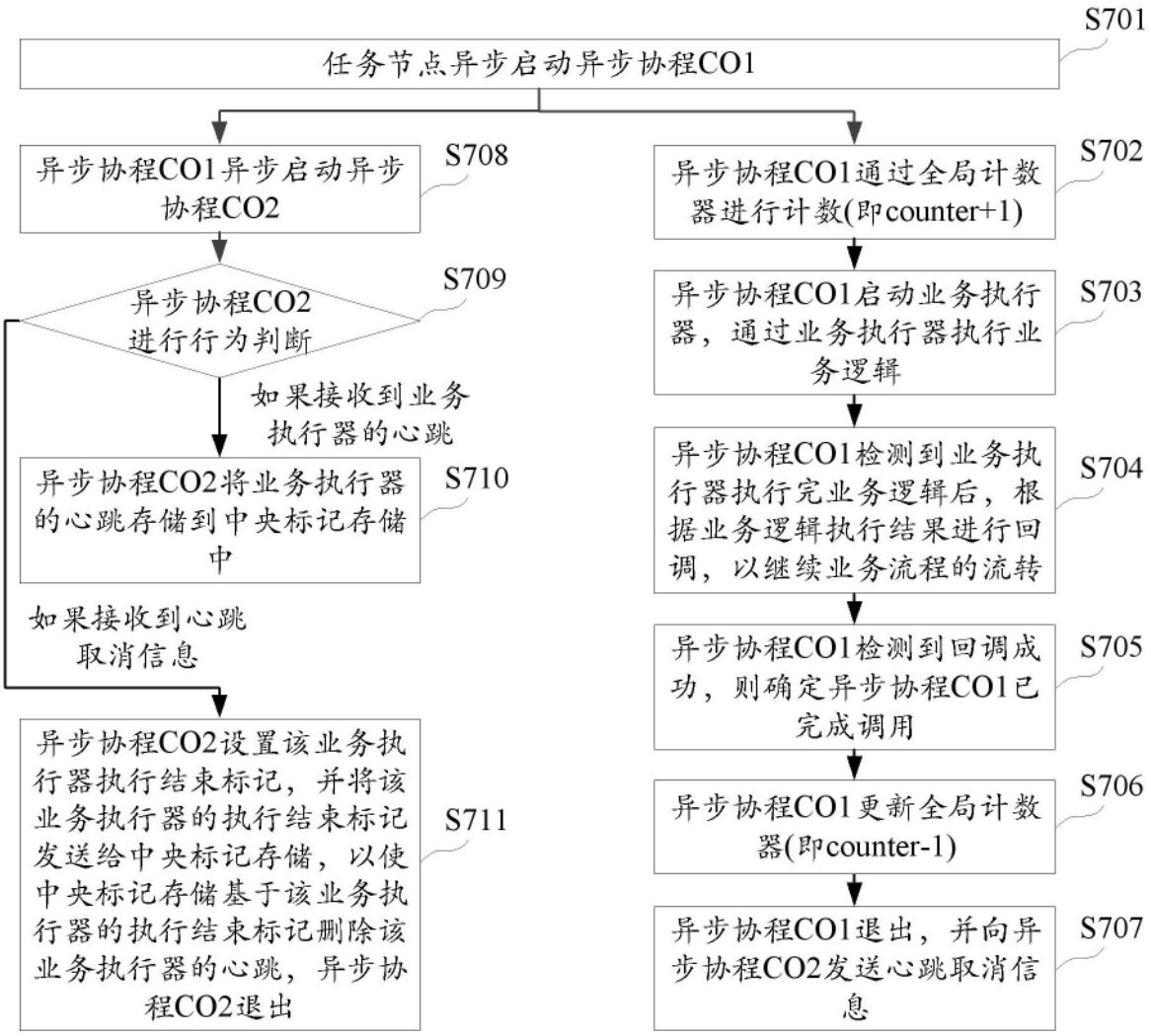

S701
任务节点异步启动异步协程CO1
异步协程CO1异步启动异步协程CO2
S708
异步协程CO1通过全局计数器进行计数(即counter+1)
S702
异步协程CO2进行行为判断
S709
异步协程CO1启动业务执行器，通过业务执行器执行业务逻辑
S703
如果接收到业务执行器的心跳
异步协程CO2将业务执行器的心跳存储到中央标记存储中
S710
异步协程CO1检测到业务执行器执行完业务逻辑后，根据业务逻辑执行结果进行回调，以继续业务流程的流转
S704
如果接收到心跳取消信息
异步协程CO1检测到回调成功，则确定异步协程CO1已完成调用
S705
异步协程CO2设置该业务执行器执行结束标记，并将该业务执行器的执行结束标记发送给中央标记存储，以使中央标记存储基于该业务执行器的执行结束标记删除该业务执行器的心跳，异步协程CO2退出
S711
异步协程CO1更新全局计数器(即counter-1)
S706
异步协程CO1退出，并向异步协程CO2发送心跳取消信息
S707

图7

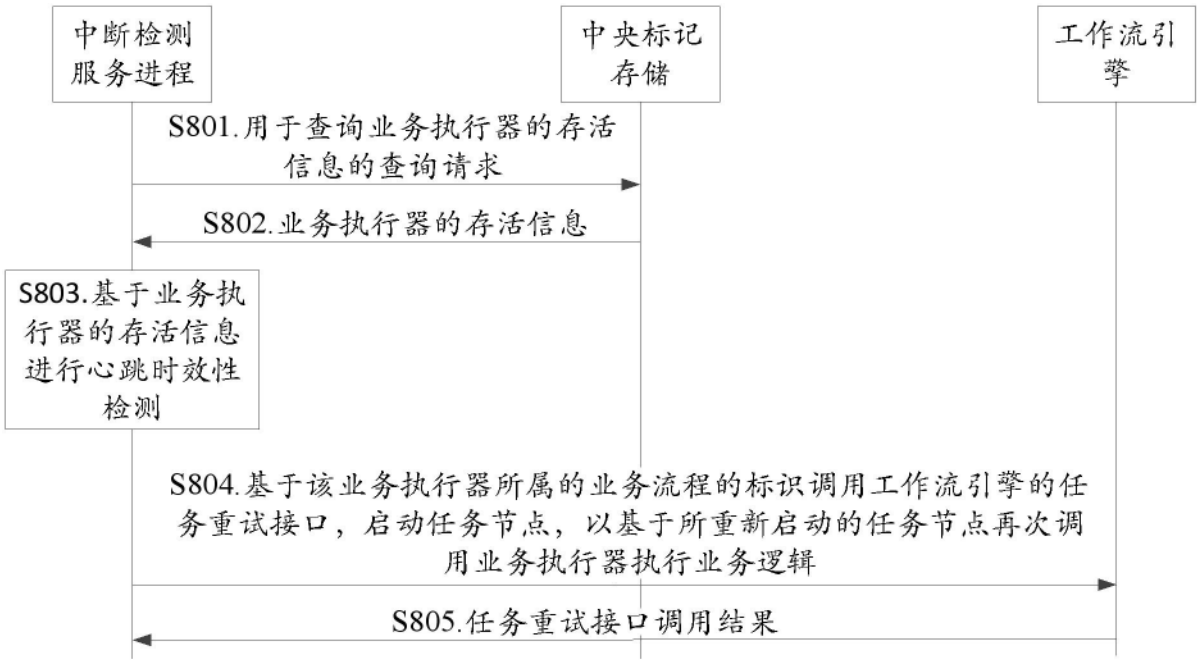

中断检测服务进程
中央标记存储
工作流引擎
S801.用于查询业务执行器的存活信息的查询请求
S802.业务执行器的存活信息
S803.基于业务执行器的存活信息进行心跳时效性检测
S804.基于该业务执行器所属的业务流程的标识调用工作流引擎的任务重试接口，启动任务节点，以基于所重新启动的任务节点再次调用业务执行器执行业务逻辑
S805.任务重试接口调用结果

图8

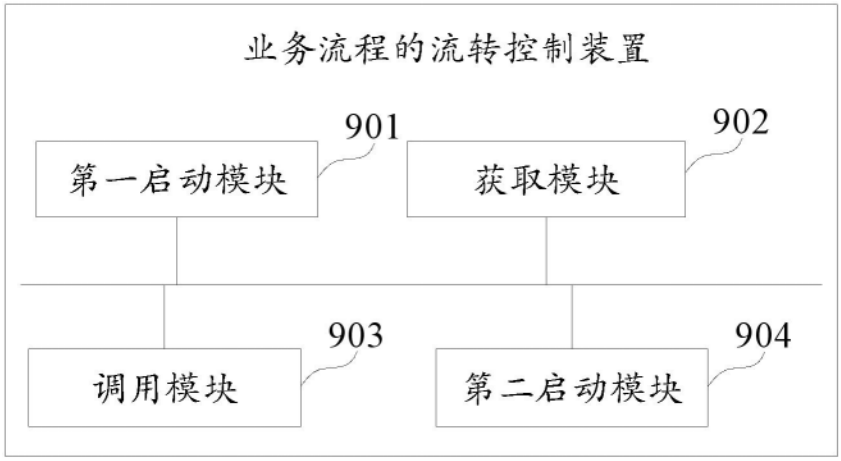

业务流程的流转控制装置
901
第一启动模块
902
获取模块
903
调用模块
904
第二启动模块

图9

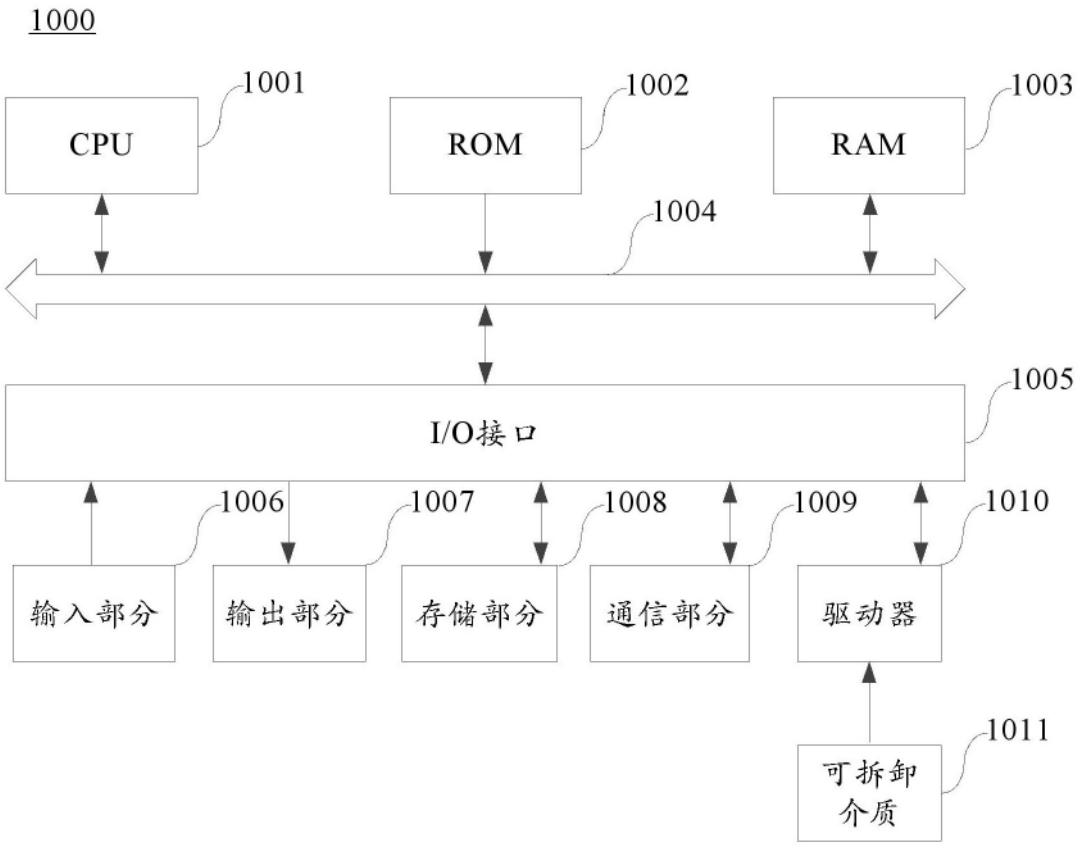

1000
CPU
1001
ROM
1002
RAM
1003
1004
I/O接口
1005
输入部分
1006
输出部分
1007
存储部分
1008
通信部分
1009
驱动器
1010
可拆卸介质
1011

图10