



(12) 发明专利申请

(10) 申请公布号 CN 115604267 A

(43) 申请公布日 2023. 01. 13

(21) 申请号 202211172237.9

(22) 申请日 2022.09.26

(71) 申请人 深圳市欧瑞博科技股份有限公司
地址 518051 广东省深圳市南山区学苑大道1001号南山智园A7栋7楼

(72) 发明人 夏文

(74) 专利代理机构 华进联合专利商标代理有限公司 44224
专利代理师 左帮胜

(51) Int. Cl.

H04L 67/1001 (2022.01)

H04L 67/1008 (2022.01)

H04L 67/562 (2022.01)

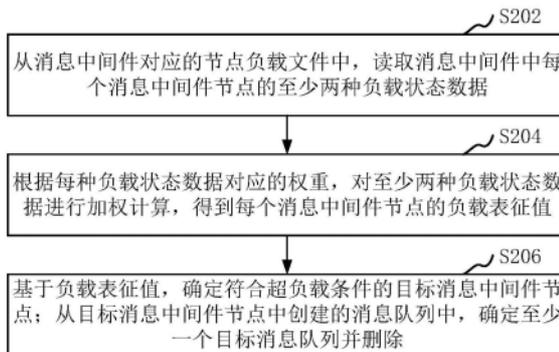
权利要求书2页 说明书12页 附图4页

(54) 发明名称

针对消息中间件的负载均衡方法、装置、设备和存储介质

(57) 摘要

本申请涉及一种针对消息中间件的负载均衡方法、装置、设备和存储介质。所述方法包括：从消息中间件对应的节点负载文件中，读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据；所述节点负载文件是通过执行脚本而将从消息中间件中获取的各消息中间件节点的负载状态数据写入对应的文件中生成的；根据每种所述负载状态数据对应的权重，对所述至少两种负载状态数据进行加权计算，得到每个消息中间件节点的负载表征值；基于所述负载表征值，确定符合超负载条件的目标消息中间件节点；从所述目标消息中间件节点中创建的消息队列中，确定至少一个目标消息队列并删除。采用本方法能够提高消息中间件的适应性。



1. 一种针对消息中间件的负载均衡方法,其特征在于,所述方法包括:

从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据;所述节点负载文件是通过执行脚本而将从消息中间件中获取的各消息中间件节点的负载状态数据写入对应的文件中生成的;

根据每种所述负载状态数据对应的权重,对所述至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值;

基于所述负载表征值,确定符合超负载条件的目标消息中间件节点;

从所述目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。

2. 根据权利要求1所述的方法,其特征在于,所述从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据包括:

通过动态调整服务,按照预设读取时间段,从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据;

其中,所述动态调整服务是用于对消息队列进行动态删除的服务应用。

3. 根据权利要求1所述的方法,其特征在于,所述负载状态数据包括消息中间件节点的处理器的资源占用数据和待处理任务数据;所述根据每种所述负载状态数据对应的权重,对所述至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值包括:

基于消息中间件节点对应的任务类型,确定所述资源占用数据对应的第一权重和所述待处理任务数据对应的第二权重;

根据所述第一权重和所述第二权重对所述资源占用数据和所述待处理任务数据进行加权计算,得到每个消息中间件节点的负载表征值。

4. 根据权利要求3所述的方法,其特征在于,在所述从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据之前,还包括:

通过执行脚本,按照预设写入时间段,从消息中间件中获取每个消息中间件节点的处理器的资源占用数据和待处理任务数据,得到每个消息中间件节点的负载状态数据;

通过执行脚本,将每个消息中间件节点的负载状态数据写入节点负载文件。

5. 根据权利要求1所述的方法,其特征在于,所述基于所述负载表征值,确定符合超负载条件的目标消息中间件节点包括:

基于每个消息中间件节点的负载表征值,确定超负载阈值;

确定所述负载表征值达到超负载阈值的目标消息中间件节点。

6. 根据权利要求1所述的方法,其特征在于,所述从所述目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除包括:

基于预先配置的可删除队列集合,从所述目标消息中间件节点中创建的消息队列中选出至少一个目标消息队列;选取的所述目标消息队列与所述可删除队列集合中的队列元素相匹配;

删除所述至少一个目标消息队列。

7. 根据权利要求1至6任意一项所述的方法,其特征在于,所述目标消息队列是由目标业务服务应用创建的;所述从所述目标消息中间件节点中创建的消息队列中,确定至少一

个目标消息队列并删除之后,还包括:

在所述目标消息队列被删除的情况下,通过所述目标业务服务应用随机重建目标消息队列至消息中间件中的任意消息中间件节点。

8. 一种针对消息中间件的负载均衡装置,其特征在于,所述装置包括:

获取模块,用于从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据;所述节点负载文件是通过执行脚本而将从消息中间件中获取各消息中间件节点的负载状态数据写入对应的文件中生成的;

计算模块,用于根据每种所述负载状态数据对应的权重,对所述至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值;

确定模块,用于基于所述负载表征值,确定符合超负载条件的目标消息中间件节点;从所述目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。

9. 一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至7中任一项所述的方法的步骤。

10. 一种计算机可读存储介质,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权利要求1至7中任一项所述的方法的步骤。

针对消息中间件的负载均衡方法、装置、设备和存储介质

技术领域

[0001] 本申请涉及消息传输技术领域，特别是涉及一种针对消息中间件的负载均衡方法、装置、设备和存储介质。

背景技术

[0002] 随着消息传输技术的发展，人们提出了消息中间件的概念，消息中间件能够处理网络通信，当网络连接不可用，消息被暂存于队列当中，当网络畅通的时候再将消息转发给相应的应用程序或者服务，可以在分布式系统中实现异步通信。

[0003] 传统技术中，消息中间件无法实现自适应负载均衡，如果数据量较大的业务数据都创建在了消息中间件集群中的某一个消息中间件节点，那么会导致这个消息中间件节点负载较高，导致消息中间件节点负载不均衡的问题。

发明内容

[0004] 基于此，有必要针对上述技术问题，提供一种能够提高适应性的针对消息中间件的负载均衡方法、装置、计算机设备、计算机可读存储介质和计算机程序产品。

[0005] 第一方面，本申请提供了一种针对消息中间件的负载均衡方法。所述方法包括：

[0006] 从消息中间件对应的节点负载文件中，读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据；所述节点负载文件是通过执行脚本而将从消息中间件中获取的各消息中间件节点的负载状态数据写入对应的文件中生成的；

[0007] 根据每种所述负载状态数据对应的权重，对所述至少两种负载状态数据进行加权计算，得到每个消息中间件节点的负载表征值；

[0008] 基于所述负载表征值，确定符合超负载条件的目标消息中间件节点；

[0009] 从所述目标消息中间件节点中创建的消息队列中，确定至少一个目标消息队列并删除。

[0010] 在其中一个实施例中，所述从消息中间件对应的节点负载文件中，读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据包括：

[0011] 通过动态调整服务，按照预设读取时间段，从消息中间件对应的节点负载文件中，读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据；

[0012] 其中，所述动态调整服务是用于对消息队列进行动态删除的服务应用。

[0013] 在其中一个实施例中，所述负载状态数据包括消息中间件节点的处理器的资源占用数据和待处理任务数据；所述根据每种所述负载状态数据对应的权重，对所述至少两种负载状态数据进行加权计算，得到每个消息中间件节点的负载表征值包括：

[0014] 基于消息中间件节点对应的任务类型，确定所述资源占用数据对应的第一权重和所述待处理任务数据对应的第二权重；

[0015] 根据所述第一权重和所述第二权重对所述资源占用数据和所述待处理任务数据进行加权计算，得到每个消息中间件节点的负载表征值。

[0016] 在其中一个实施例中,在所述从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据之前,还包括:

[0017] 通过执行脚本,按照预设写入时间段,从消息中间件中获取每个消息中间件节点的处理器的资源占用数据和待处理任务数据,得到每个消息中间件节点的负载状态数据;

[0018] 通过执行脚本,将每个消息中间件节点的负载状态数据写入节点负载文件。

[0019] 在其中一个实施例中,所述基于所述负载表征值,确定符合超负载条件的目标消息中间件节点包括:

[0020] 基于每个消息中间件节点的负载表征值,确定超负载阈值;

[0021] 确定所述负载表征值达到超负载阈值的目标消息中间件节点。

[0022] 在其中一个实施例中,所述从所述目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除包括:

[0023] 基于预先配置的可删除队列集合,从所述目标消息中间件节点中创建的消息队列中选出至少一个目标消息队列;选取的所述目标消息队列与所述可删除队列集合中的队列元素相匹配;

[0024] 删除所述至少一个目标消息队列。

[0025] 在其中一个实施例中,所述目标消息队列是由目标业务服务应用创建的;所述从所述目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除之后,还包括:

[0026] 在所述目标消息队列被删除的情况下,通过所述目标业务服务应用随机重建目标消息队列至消息中间件中的任意消息中间件节点。

[0027] 第二方面,本申请还提供了一种针对消息中间件的负载均衡装置。所述装置包括:

[0028] 获取模块,用于从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据;所述节点负载文件是通过执行脚本而将从消息中间件中获取各消息中间件节点的负载状态数据写入对应的文件中生成的;

[0029] 计算模块,用于根据每种所述负载状态数据对应的权重,对所述至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值;

[0030] 确定模块,用于基于所述负载表征值,确定符合超负载条件的目标消息中间件节点;从所述目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。

[0031] 在其中一个实施例中,所述获取模块,用于通过动态调整服务,按照预设读取时间段,从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据;其中,所述动态调整服务是用于对消息队列进行动态删除的服务应用。

[0032] 在其中一个实施例中,所述负载状态数据包括消息中间件节点的处理器的资源占用数据和待处理任务数据;所述计算模块,还用于基于消息中间件节点对应的任务类型,确定所述资源占用数据对应的第一权重和所述待处理任务数据对应的第二权重;根据所述第一权重和所述第二权重对所述资源占用数据和所述待处理任务数据进行加权计算,得到每个消息中间件节点的负载表征值。

[0033] 在其中一个实施例中,所述获取模块,还用于通过执行脚本,按照预设写入时间段,从消息中间件中获取每个消息中间件节点的处理器的资源占用数据和待处理任务数

据,得到每个消息中间件节点的负载状态数据;通过执行脚本,将每个消息中间件节点的负载状态数据写入节点负载文件。

[0034] 在其中一个实施例中,所述确定模块,用于基于每个消息中间件节点的负载表征值,确定超负载阈值;确定所述负载表征值达到超负载阈值的消息中间件节点。

[0035] 在其中一个实施例中,所述确定模块,用于基于预先配置的可删除队列集合,从所述消息中间件节点中创建的消息队列中选出至少一个目标消息队列;选取的所述目标消息队列与所述可删除队列集合中的队列元素相匹配;删除所述至少一个目标消息队列。

[0036] 在其中一个实施例中,所述目标消息队列是由目标业务服务应用创建的;所述确定模块,用于在所述目标消息队列被删除的情况下,通过所述目标业务服务应用随机重建目标消息队列至消息中间件中的任意消息中间件节点。

[0037] 第三方面,本申请还提供了一种计算机设备。所述计算机设备包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现本申请所述方法各实施例中的步骤。

[0038] 第四方面,本申请还提供了一种计算机可读存储介质。所述计算机可读存储介质,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本申请所述方法各实施例中的步骤。

[0039] 第五方面,本申请还提供了一种计算机程序产品。所述计算机程序产品,包括计算机程序,该计算机程序被处理器执行时实现本申请所述方法各实施例中的步骤。

[0040] 上述针对消息中间件的负载均衡方法、装置、计算机设备、存储介质和计算机程序产品,从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的至少两种负载状态数据;节点负载文件是通过执行脚本而将从消息中间件中获取的各消息中间件节点的负载状态数据写入对应的文件中生成的;根据每种负载状态数据对应的权重,对至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值;基于负载表征值,确定符合超负载条件的目标消息中间件节点;从目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。通过负载状态数据和相应的权重计算每个中间件节点的负载表征值,后续在确定符合超负载条件的目标消息中间件节点之后,删除目标消息队列以降低负载,从而能够自适应地调整消息中间件节点的负载,提高了适应性。

附图说明

[0041] 图1为一个实施例中针对消息中间件的负载均衡方法的应用环境图;

[0042] 图2为一个实施例中针对消息中间件的负载均衡方法的流程示意图;

[0043] 图3为一个实施例中针对消息中间件的负载均衡方法的架构图;

[0044] 图4为一个实施例中针对消息中间件的负载均衡装置的结构框图;

[0045] 图5为一个实施例中计算机设备的内部结构图;

[0046] 图6为另一个实施例中计算机设备的内部结构图。

具体实施方式

[0047] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对

本申请进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0048] 本申请实施例提供的针对消息中间件的负载均衡方法,可以应用于如图1所示的应用环境中。其中,消息中间件102通过网络与负载调整服务器104进行通信。负载调整服务器104可以通过执行脚本而将从消息中间件中获取的各消息中间件节点的负载状态数据,并将负载状态数据写入对应的文件中以生成消息中间件102对应的节点负载文件。负载调整服务器104可以从消息中间件102对应的节点负载文件中,读取消息中间件102中每个消息中间件节点的至少两种负载状态数据。负载调整服务器104可以根据每种负载状态数据对应的权重,对至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值。负载调整服务器104可以基于负载表征值,确定符合超负载条件的目标消息中间件节点。负载调整服务器104可以从目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。其中,消息中间件102与负载调整服务器104均可以用独立的服务器或者是多个服务器组成的服务器集群来实现。

[0049] 在一个实施例中,消息中间件102可以由多个服务器组成的服务器集群实现。消息中间件包括的每个消息中间件节点可以涉及一个或多个服务器。其中,消息中间件是基于队列与消息传递技术,用于支持同步或异步、可靠的消息传输。每个消息中间件节点在与业务服务应用建立连接之后,可以支持业务服务应用进行消息传输,即,消息中间件节点用于与业务服务应用建立连接,以使得业务服务应用在消息中间件节点中创建消息队列。业务服务应用是用于支持部署在终端的业务应用的服务程序。可以理解,终端中的业务应用响应用户的操作,生成相应的消息,通过部署在应用服务器中的业务服务应用将上述消息发送给所连接的消息中间件节点。

[0050] 在一个实施例中,在智能家居的场景下,智能中控设备中部署有业务应用,智能中控设备可以响应针对家居设备的控制操作,通过业务应用创建控制指令。智能中控设备可以通过网络与应用服务器进行通信,将控制指令发送至应用服务器。应用服务器可以通过业务服务应用将控制指令创建至所连接的目标中间件节点的消息队列中。

[0051] 在一个实施例中,业务服务应用可以部署在终端,终端可以通过网络与消息中间件通信。终端可以但不限于各种个人计算机、笔记本电脑、智能手机、平板电脑、物联网设备和便携式可穿戴设备,物联网设备可为智能音箱、智能电视、智能空调、智能车载设备等。便携式可穿戴设备可为智能手表、智能手环、头戴设备等。

[0052] 在一个实施例中,如图2所示,提供了一种针对消息中间件的负载均衡方法,以该方法应用于图1中的负载调整服务器为例进行说明,包括以下步骤:

[0053] S202,从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的至少两种负载状态数据。

[0054] 其中,节点负载文件是通过执行脚本而将从消息中间件中获取的各消息中间件节点的负载状态数据写入对应的文件中生成的。负载状态数据用于指示消息中间件节点的负载状态。可以理解,多种负载状态数据能够从多个维度体现消息中间件节点的负载状态。比如,一种负载状态数据体现了消息中间件节点的繁忙程度,另一种负载状态数据体现了消息中间件节点的负载量。

[0055] 具体地,负载调整服务器可以通过执行脚本从消息中间件获取各消息中间件节点

的负载状态数据,并将获取到的负载状态数据写入至文件中得到消息中间件对应的节点负载文件。负载调整服务器可以从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的多种负载状态数据。

[0056] 在一个实施例中,每个消息中间件节点可以单独运行在一个可运行应用程序的容器,也可以单独运行在一台终端或服务器,也可以单独运行在一个云服务的虚拟主机。

[0057] S204,根据每种负载状态数据对应的权重,对至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值。

[0058] 其中,负载表征值用于表征消息中间件节点的整体负载情况。可以理解,每种负载状态数据只能表征单一维度下的负载情况,结合至少两种负载状态数据得到的负载表征值能够综合全面地体现消息中间件节点的负载情况。

[0059] 具体地,负载调整服务器中预设每种负载状态数据对应的权重。负载调整服务器可以使用每种负载状态数据对应的权重进行加权求和计算,得到每个消息中间件节点的负载表征值。可以理解,负载调整服务器可以使用每种负载状态数据对应的权重分别进行加权,得到加权后的每种负载状态数据。负载调整服务器可以通过计算加权后的多种负载状态数据进行求和计算,得到每个中间件节点的负载表征值。

[0060] S206,基于负载表征值,确定符合超负载条件的目标消息中间件节点;从目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。

[0061] 具体地,消息中间件包括多个消息中间件节点。负载调整服务器可以确定多个消息中间件节点的负载表征值的平均值,得到消息中间件对应的负载平均值。负载调整服务器可以基于负载平均值,确定符合超负载条件。负载调整服务器可以从多个消息中间件节点中确定符合超负载条件的目标消息中间件节点,然后从目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。

[0062] 在一个实施例中,超负载条件可以是负载表征值超出负载平均值的固定比例。可以理解,负载表征值超出负载平均值的固定比例的情况下,相应的消息中间件节点可以作为目标中间件节点。比如,固定比例可以是30%,负载表征值超出负载平均值的30%,即,负载表征值大于负载平均值的1.3倍的情况下,相应的消息中间件节点可以作为目标中间件节点。

[0063] 上述针对消息中间件的负载均衡方法中,从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的至少两种负载状态数据;节点负载文件是通过执行脚本而将从消息中间件中获取的各消息中间件节点的负载状态数据写入对应的文件中生成的;根据每种负载状态数据对应的权重,对至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值;基于负载表征值,确定符合超负载条件的目标消息中间件节点;从目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。通过负载状态数据和相应的权重计算每个中间件节点的负载表征值,后续在确定符合超负载条件的目标消息中间件节点之后,删除目标消息队列以降低负载,从而能够自适应地调整消息中间件节点的负载,提高了适应性。

[0064] 在一个实施例中,从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的至少两种负载状态数据的步骤包括:通过动态调整服务,按照预设读取时间段,从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的

至少两种负载状态数据;其中,动态调整服务是用于对消息队列进行动态删除的服务应用。

[0065] 其中,预设读取时间段是预先设置的读取负载状态数据的固定时间间隔,比如,预设读取时间段可以是2秒。

[0066] 具体地,负载调整服务器中部署有动态调整服务。负载调整服务器可以通过动态调整服务,按照预设读取时间段,即,每隔一预设读取时间段从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的多种负载状态数据。后续负载调整服务器可以将读取到的负载状态数据写入节点负载文件中。

[0067] 在一个实施例中,负载调整服务器可以按照预设写入时间段将读取到的负载状态数据写入至节点负载文件中。其中,预设写入时间段是预先设置的将负载状态数据写入文件的固定时间间隔。预设写入时间段和预设读取时间段可以是一致的。

[0068] 在一个实施例中,预设写入时间段可以小于或等于预设读取时间段,以保证负载调整服务器能够读取到新的负载状态数据。

[0069] 本实施例中,通过动态调整服务,按照预设读取时间段,从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的至少两种负载状态数据,后续能够基于读取到的负载状态数据针对消息中间件进行负载均衡,提高了适应性。

[0070] 在一个实施例中,负载状态数据包括消息中间件节点的处理器资源占用数据和待处理任务数据;根据每种负载状态数据对应的权重,对至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值包括:基于消息中间件节点对应的任务类型,确定资源占用数据对应的第一权重和待处理任务数据对应的第二权重;根据第一权重和第二权重对资源占用数据和待处理任务数据进行加权计算,得到每个消息中间件节点的负载表征值。

[0071] 其中,资源占用数据用于指示消息中间件节点的繁忙程度。待处理任务数据用于指示消息中间件节点的负载数量。可以理解,消息中间件节点过于繁忙或者负载数量过高会导致该消息中间件节点崩溃,无法继续支持消息传输。

[0072] 其中,任务类型是消息中间件节点处理器处理的任务的类型。比如,任务类型可以包括计算密集型(CPU-bound)和存取密集型(I/O-bound)。计算密集型任务的特点是要进行大量的计算,消耗处理器的计算资源,因此处理计算密集型任务资源占用数据影响更大,待处理任务数据影响更小。存取密集型任务的特点是计算资源消耗很少,任务的大部分时间都在等待存取操作(I/O操作)完成,因此处理存取密集型任务,任务资源占用数据影响更小,待处理任务数据影响更大。

[0073] 具体地,负载调整服务器中预先针对不同的任务类型以及不同的负载状态数据设置了不同的权重。可以理解,对应于计算密集型的待处理任务数据对应的第二权重相较于存取密集型设置的更高,对应于存取密集型的资源占用数据对应的第二权重相较于计算密集型设置的更高。负载调整服务器可以确定与消息中间件节点的任务类型相匹配的资源占用数据对应的第一权重和待处理任务数据对应的第二权重。负载调整服务器可以使用第一权重和第二权重对资源占用数据和待处理任务数据进行加权求和计算,得到每个消息中间件节点的负载表征值。可以理解,负载调整服务器可以使用第一权重对资源占用数据进行加权,得到加权后的资源占用数据,以及使用第二权重对待处理任务数据进行加权,得到加权后的待处理任务数据。负载调整服务器可以对加权后的资源占用数据和加权后的待处理

任务数据进行求和,得到消息中间件节点的负载表征值。

[0074] 本实施例中,基于消息中间件节点对应的任务类型,确定资源占用数据对应的第一权重和待处理任务数据对应的第二权重;根据第一权重和第二权重对资源占用数据和待处理任务数据进行加权计算,得到每个消息中间件节点的负载表征值,后续能够基于负载表征值确定目标消息中间件节点,通过对目标消息中间件节点的目标消息队列进行删除,实现对消息中间件的负载均衡。

[0075] 在一个实施例中,在从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的至少两种负载状态数据之前,还包括:通过执行脚本,按照预设写入时间段,从消息中间件中获取每个消息中间件节点的处理器资源占用数据和待处理任务数据,得到每个消息中间件节点的负载状态数据;通过执行脚本,将每个消息中间件节点的负载状态数据写入节点负载文件。

[0076] 具体地,资源占用数据可以包括消息中间件节点的处理器使用率(CPUrate)。待处理任务数据可以包括消息中间件的处理器负载量(CPUload)。负载调整服务器可以通过执行脚本,按照预设写入时间段,即,每隔一预设写入时间段从消息中间件中获取每个消息中间件节点的处理器使用率和处理器负载量,得到每个消息中间件节点的负载状态数据。负载调整服务器可以通过执行脚本,将每个消息中间件节点的负载状态数据写入节点负载文件。可以理解,负载调整服务器可以读取每个消息中间件节点的两种负载状态数据,这两种负载状态数据可以是处理器使用率和处理器负载量。

[0077] 其中,处理器使用率表示处理器的实际工作量。处理器使用率可以指示消息中间件节点的繁忙情况。处理器负载量表示等待使用处理器的任务数量。处理器负载量可以指示消息中间件节点的负载数量。

[0078] 在一个实施例中,负载表征值=处理器使用率*第一权重+处理器负载量*第二权重。

[0079] 本实施例中,通过执行脚本,按照预设写入时间段,从消息中间件中获取每个消息中间件节点的处理器资源占用数据和待处理任务数据,得到每个消息中间件节点的负载状态数据;通过执行脚本,将每个消息中间件节点的负载状态数据写入节点负载文件,后续能够基于节点负载文件中的负载状态数据针对消息中间件进行负载均衡。

[0080] 在一个实施例中,基于负载表征值,确定符合超负载条件的目标消息中间件节点的步骤包括:基于每个消息中间件节点的负载表征值,确定超负载阈值;确定负载表征值达到超负载阈值的消息中间件节点。

[0081] 其中,超负载阈值是用于判定是否超负载的阈值。

[0082] 具体地,超负载阈值可以等于固定倍数的负载平均值。负载调整服务器可以从多个消息中间件节点中确定负载表征值最大的消息中间件节点。负载调整服务器可以对最大的负载表征值和负载平均值进行求差计算,得到负载差值。在负载差值与负载平均值的比值大于固定比例的情况下,确定最大负载表征值达到超负载阈值,将相应的消息中间件节点作为目标消息中间件节点。其中,固定比例与固定倍数相匹配,固定比例等于固定倍数减去1。比如固定比例是30%,超负载条件可以是(最大的负载表征值-负载平均值)/负载平均值>30%。

[0083] 在一个实施例中,对于多个消息中间件节点的负载表征值达到超负载阈值的情

况,每次调整最大负载表征值的那个消息中间件节点,直到该消息中间件节点的负载表征值不再最大,下次调整就会选取另外的消息中间件节点(最大负载表征值对应的消息中间件节点)进行调整,从而使得负载表征值超过负载平均值30%的消息中间件节点的负载都会调整下来。

[0084] 本实施例中,基于每个消息中间件节点的负载表征值,确定超负载阈值;确定负载表征值达到超负载阈值的消息中间件节点,能够不断地对超负载的消息中间件节点进行调整,实现消息中间件的负载均衡,提高适应性。

[0085] 在一个实施例中,从目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除的步骤包括:基于预先配置的可删除队列集合,从目标消息中间件节点中创建的消息队列中选出至少一个目标消息队列;选取的目标消息队列与可删除队列集合中的队列元素相匹配;删除至少一个目标消息队列。

[0086] 其中,可删除队列集合中包括对消息中间件节点影响较大的消息队列相关信息。可以理解,可删除队列集合的队列元素可以对应消息队列的属性。比如队列元素可以指示进出速度大于预设最大速度的消息队列。可以理解,进出速度大于1000次/秒会对消息中间件的负载影响较大。

[0087] 具体地,负载调整服务器可以确定预先配置的可删除队列集合,从目标消息中间件节点中创建的消息队列中选出与可删除队列集合中的队列元素相匹配的目标消息队列,并删除目标消息队列。

[0088] 在一个实施例中,在存在多个消息队列分别与多个队列元素相匹配的情况下,负载调整服务器可以任意选取其中的一个消息队列作为目标消息队列。可以理解,可删除队列集合中指示的消息队列都是对负载影响较大的消息队列,可以随机选取一个,随机选取一个要优于选取消息量最大的消息队列。如果直接选取删除消息量最大的,可能导致该消息中间件节点负载降低过快,又导致不均衡。

[0089] 在一个实施例中,存在删除消息队列会导致消息丢失的情况,因此会丢失消息的消息队列不可记录至可删除队列集合中,人工可以根据具体情况修改可删除队列集合。

[0090] 在一个实施例中,负载调整服务器可以与应用服务器通信,确定目标业务服务应用创建的原始消息队列,在目标消息队列与原始消息队列相符的情况下,对目标消息队列进行删除。可以理解,目标消息队列与原始消息队列相符表明目标消息队列不存在丢消息的情况。

[0091] 在一个实施例中,在目标消息队列与原始消息队列不相符的情况下,负载调整服务器可以重新确定目标消息队列。

[0092] 本实施例中,基于预先配置的可删除队列集合,从目标消息中间件节点中创建的消息队列中选出至少一个目标消息队列;选取的目标消息队列与可删除队列集合中的队列元素相匹配;删除至少一个目标消息队列,实现了消息中间件的负载均衡,提高了适应性。

[0093] 在一个实施例中,目标消息队列是由目标业务服务应用创建的;从目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除之后,还包括:在目标消息队列被删除的情况下,通过目标业务服务应用随机重建目标消息队列至消息中间件中的任意消息中间件节点。

[0094] 具体地,在目标消息队列被删除的情况下,目标业务服务应用所在的应用服务器

可以通过监听消息中间件获知目标消息队列被删除的事件,应用服务器可以通过目标业务服务应用随机重建目标消息队列至消息中间件中的任意消息中间件节点。

[0095] 在一个实施例中,业务服务应用存在多个,部署于多个应用服务器组成的集群中。在启动业务应用服务的情况下,应用服务器会使用轮询的方式与消息中间件建立连接,因此业务服务应用所连接的消息中间件节点是不固定的。在监听到目标消息队列被删除的时间之后,目标业务服务应用所在的应用服务器通过与消息中间件的连接,重新将目标消息队列创建至任意消息中间件节点。

[0096] 在一个实施例中,通过先连接至消息中间件的应用服务器将目标消息队列重建至任意消息中间件节点。可以理解,多个应用服务器组成的集群可以监听到目标消息队列被删除的时间,多个应用服务器均可以重建目标消息队列,因此通过先连接到消息中间件的应用服务器进行重建。

[0097] 本实施例中,在目标消息队列被删除的情况下,通过目标业务服务应用随机重建目标消息队列至消息中间件中的任意消息中间件节点,在实现负载均衡的同时尽可能地支持消息的传输,提高了适应性。

[0098] 在一个实施例中,如图3所示提供了针对消息中间件的负载均衡方法的架构图。负载调整服务器可以通过执行脚本每隔一预设写入时间段获取消息中间件包括的每个消息中间件节点的处理器使用率和处理器负载量。其中,预设写入时间段可以是2秒,脚本可以是shell脚本。负载调整服务器可以通过执行脚本将获取到的处理器使用率和处理器负载量写入至节点负载文件。节点负载文件中包括消息中间件中每个消息中间件节点的处理器使用率和处理器负载量。

[0099] 负载调整服务器可以通过动态调整服务每隔一预设读取时间段从节点负载文件中读取每个消息中间件节点的处理器使用率和处理器负载量,并对处理器使用率和处理器负载量进行加权求和计算,得到每个消息中间件节点的负载表征值。负载调整服务器可以通过动态调整服务确定最大的负载表征值和负载平均值,并在最大的负载表征值与负载平均值之间的负载差值超过负载平均值的固定比例的情况下,将最大的负载表征值对应的消息中间件节点确定为目标中间件节点。

[0100] 负载调整服务器可以通过动态调整服务确定与可删除队列集合中的队列元素相匹配的任意一个消息队列,得到目标消息队列,并删除目标消息队列。应用服务器可以在监听到目标消息队列被删除的事件之后,通过目标业务服务应用重建目标消息队列。

[0101] 应该理解的是,虽然如上所述的各实施例所涉及的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,这些步骤可以以其它的顺序执行。而且,如上所述的各实施例所涉及的流程图中的至少一部分步骤可以包括多个步骤或者多个阶段,这些步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,这些步骤或者阶段的执行顺序也不必然是依次进行,而是可以与其它步骤或者其它步骤中的步骤或者阶段的至少一部分轮流或者交替地执行。

[0102] 基于同样的发明构思,本申请实施例还提供了一种用于实现上述所涉及的针对消息中间件的负载均衡方法的针对消息中间件的负载均衡装置。该装置所提供的解决问题的实现方案与上述方法中所记载的实现方案相似,故下面所提供的的一个或多个针对消息中间

件的负载均衡装置实施例中的具体限定可以参见上文中对于针对消息中间件的负载均衡方法的限定,在此不再赘述。

[0103] 在一个实施例中,如图4所示,提供了一种针对消息中间件的负载均衡装置400,包括:获取模块402、计算模块404和确定模块406,其中:

[0104] 获取模块402,用于从消息中间件对应的节点负载文件中,读取所述消息中间件中每个消息中间件节点的至少两种负载状态数据;所述节点负载文件是通过执行脚本而将从消息中间件中获取各消息中间件节点的负载状态数据写入对应的文件中生成的。

[0105] 计算模块404,用于根据每种所述负载状态数据对应的权重,对所述至少两种负载状态数据进行加权计算,得到每个消息中间件节点的负载表征值。

[0106] 确定模块406,用于基于所述负载表征值,确定符合超负载条件的目标消息中间件节点;从所述目标消息中间件节点中创建的消息队列中,确定至少一个目标消息队列并删除。

[0107] 在一个实施例中,获取模块402,用于通过动态调整服务,按照预设读取时间段,从消息中间件对应的节点负载文件中,读取消息中间件中每个消息中间件节点的至少两种负载状态数据;其中,动态调整服务是用于对消息队列进行动态删除的服务应用。

[0108] 在一个实施例中,负载状态数据包括消息中间件节点的处理器资源占用数据和待处理任务数据;计算模块404,还用于基于消息中间件节点对应的任务类型,确定资源占用数据对应的第一权重和待处理任务数据对应的第二权重;根据第一权重和第二权重对资源占用数据和待处理任务数据进行加权计算,得到每个消息中间件节点的负载表征值。

[0109] 在一个实施例中,获取模块402,还用于通过执行脚本,按照预设写入时间段,从消息中间件中获取每个消息中间件节点的处理器资源占用数据和待处理任务数据,得到每个消息中间件节点的负载状态数据;通过执行脚本,将每个消息中间件节点的负载状态数据写入节点负载文件。

[0110] 在一个实施例中,确定模块406,用于基于每个消息中间件节点的负载表征值,确定超负载阈值;确定负载表征值达到超负载阈值的消息中间件节点。

[0111] 在一个实施例中,确定模块406,用于基于预先配置的可删除队列集合,从目标消息中间件节点中创建的消息队列中选出至少一个目标消息队列;选取的目标消息队列与可删除队列集合中的队列元素相匹配;删除至少一个目标消息队列。

[0112] 在一个实施例中,目标消息队列是由目标业务服务应用创建的;确定模块406,用于在目标消息队列被删除的情况下,通过目标业务服务应用随机重建目标消息队列至消息中间件中的任意消息中间件节点。

[0113] 上述针对消息中间件的负载均衡装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中,也可以以软件形式存储于计算机设备中的存储器中,以便于处理器调用执行以上各个模块对应的操作。

[0114] 在一个实施例中,提供了一种计算机设备,该计算机设备可以是服务器,其内部结构图可以如图5所示。该计算机设备包括处理器、存储器、输入/输出接口(Input/Output,简称I/O)和通信接口。其中,处理器、存储器和输入/输出接口通过系统总线连接,通信接口通过输入/输出接口连接到系统总线。其中,该计算机设备的处理器用于提供计算和控制能

力。该计算机设备的存储器包括非易失性存储介质和内存存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库用于存储节点负载文件。该计算机设备的输入/输出接口用于处理器与外部设备之间交换信息。该计算机设备的通信接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种针对消息中间件的负载均衡方法。

[0115] 在一个实施例中,提供了一种计算机设备,该计算机设备可以是终端,其内部结构图可以如图6所示。该计算机设备包括处理器、存储器、输入/输出接口、通信接口、显示单元和输入装置。其中,处理器、存储器和输入/输出接口通过系统总线连接,通信接口、显示单元和输入装置通过输入/输出接口连接到系统总线。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存存储器。该非易失性存储介质存储有操作系统和计算机程序。该内存存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的输入/输出接口用于处理器与外部设备之间交换信息。该计算机设备的通信接口用于与外部的终端进行有线或无线方式的通信,无线方式可通过WIFI、移动蜂窝网络、NFC(近场通信)或其他技术实现。该计算机程序被处理器执行时以实现一种针对消息中间件的负载均衡方法。该计算机设备的显示单元用于形成视觉可见的画面,可以是显示屏、投影装置或虚拟现实成像装置,显示屏可以是液晶显示屏或电子墨水显示屏,该计算机设备的输入装置可以是显示屏上覆盖的触摸层,也可以是计算机设备外壳上设置的按键、轨迹球或触控板,还可以是外接的键盘、触控板或鼠标等。

[0116] 本领域技术人员可以理解,图5和图6中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定,具体的计算机设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0117] 在一个实施例中,提供了一种计算机设备,包括存储器和处理器,存储器中存储有计算机程序,该处理器执行计算机程序时实现上述各方法实施例中的步骤。

[0118] 在一个实施例中,提供了一种计算机可读存储介质,其上存储有计算机程序,计算机程序被处理器执行时实现上述各方法实施例中的步骤。

[0119] 在一个实施例中,提供了一种计算机程序产品,包括计算机程序,该计算机程序被处理器执行时实现上述各方法实施例中的步骤。

[0120] 需要说明的是,本申请所涉及的用户信息(包括但不限于用户设备信息、用户个人信息等)和数据(包括但不限于用于分析的数据、存储的数据、展示的数据等),均为经用户授权或者经过各方充分授权的信息和数据,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。

[0121] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一非易失性计算机可读存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的各实施例中所使用的对存储器、数据库或其它介质的任何引用,均可包括非易失性和易失性存储器中的至少一种。非易失性存储器可包括只读存储器(Read-Only Memory, ROM)、磁带、软盘、闪存、光存储器、高密度嵌入式非易失性存储器、阻变存储器

(ReRAM)、磁变存储器 (Magnetoresistive Random Access Memory, MRAM)、铁电存储器 (Ferroelectric Random Access Memory, FRAM)、相变存储器 (Phase Change Memory, PCM)、石墨烯存储器等。易失性存储器可包括随机存取存储器 (Random Access Memory, RAM) 或外部高速缓冲存储器等。作为说明而非局限, RAM可以是多种形式, 比如静态随机存取存储器 (Static Random Access Memory, SRAM) 或动态随机存取存储器 (Dynamic Random Access Memory, DRAM) 等。本申请所提供的各实施例中所涉及的数据库可包括关系型数据库和非关系型数据库中至少一种。非关系型数据库可包括基于区块链的分布式数据库等, 不限于此。本申请所提供的各实施例中所涉及的处理器可为通用处理器、中央处理器、图形处理器、数字信号处理器、可编程逻辑器、基于量子计算的数据处理逻辑器等, 不限于此。

[0122] 以上实施例的各技术特征可以进行任意的组合, 为使描述简洁, 未对上述实施例中的各个技术特征所有可能的组合都进行描述, 然而, 只要这些技术特征的组合不存在矛盾, 都应当认为是本说明书记载的范围。

[0123] 以上所述实施例仅表达了本申请的几种实施方式, 其描述较为具体和详细, 但并不能因此而理解为对本申请专利范围的限制。应当指出的是, 对于本领域的普通技术人员来说, 在不脱离本申请构思的前提下, 还可以做出若干变形和改进, 这些都属于本申请的保护范围。因此, 本申请的保护范围应以所附权利要求为准。

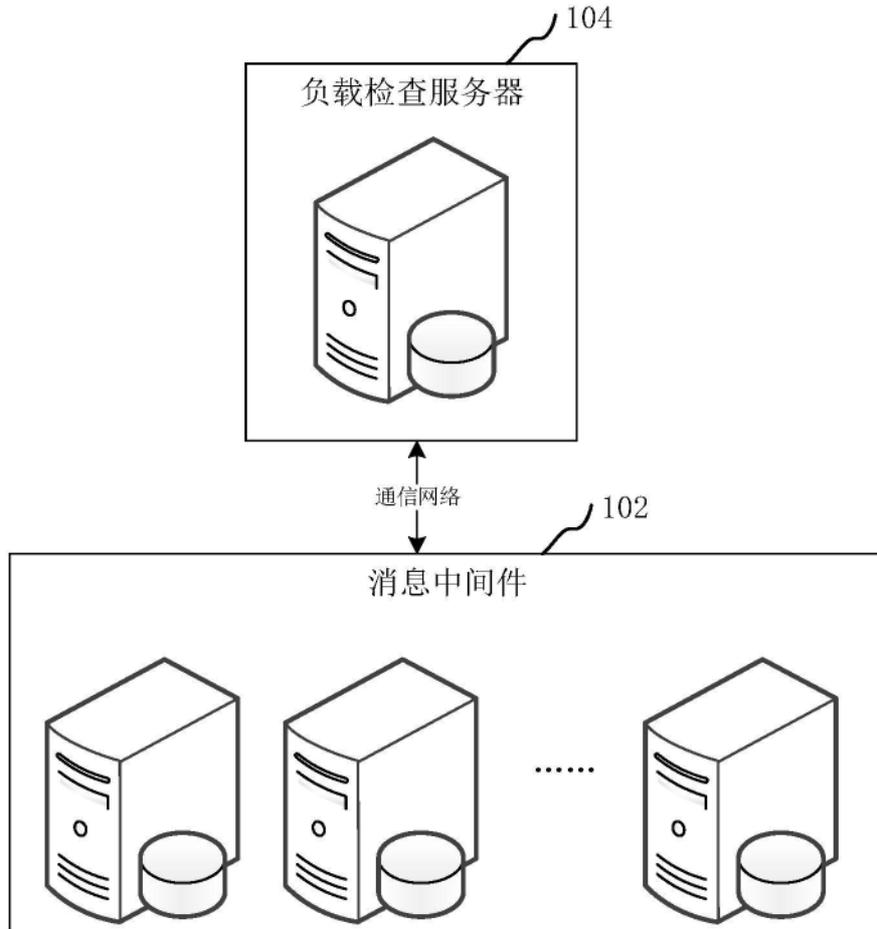


图1

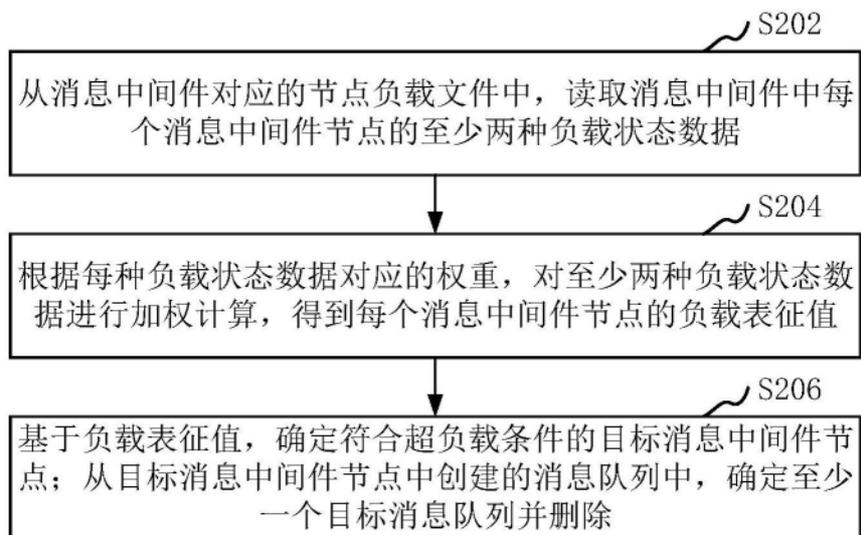


图2

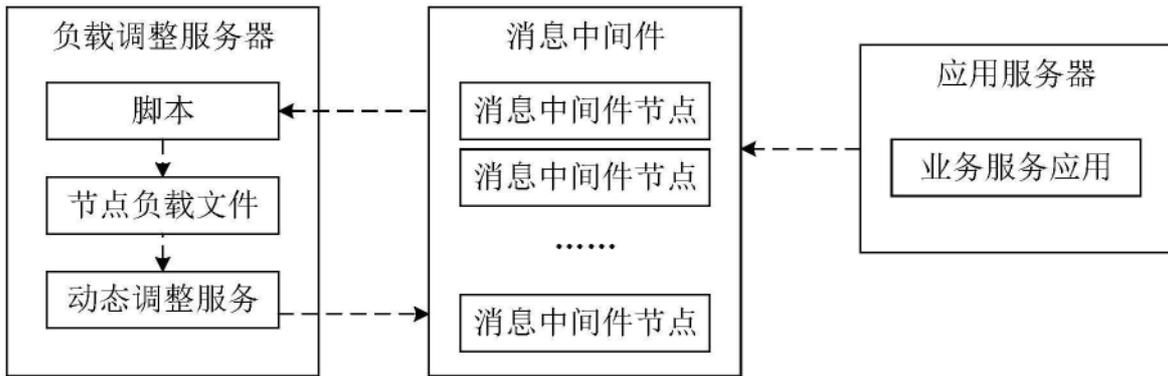


图3

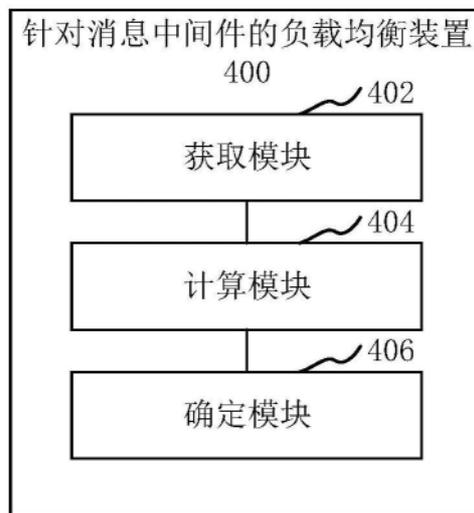


图4

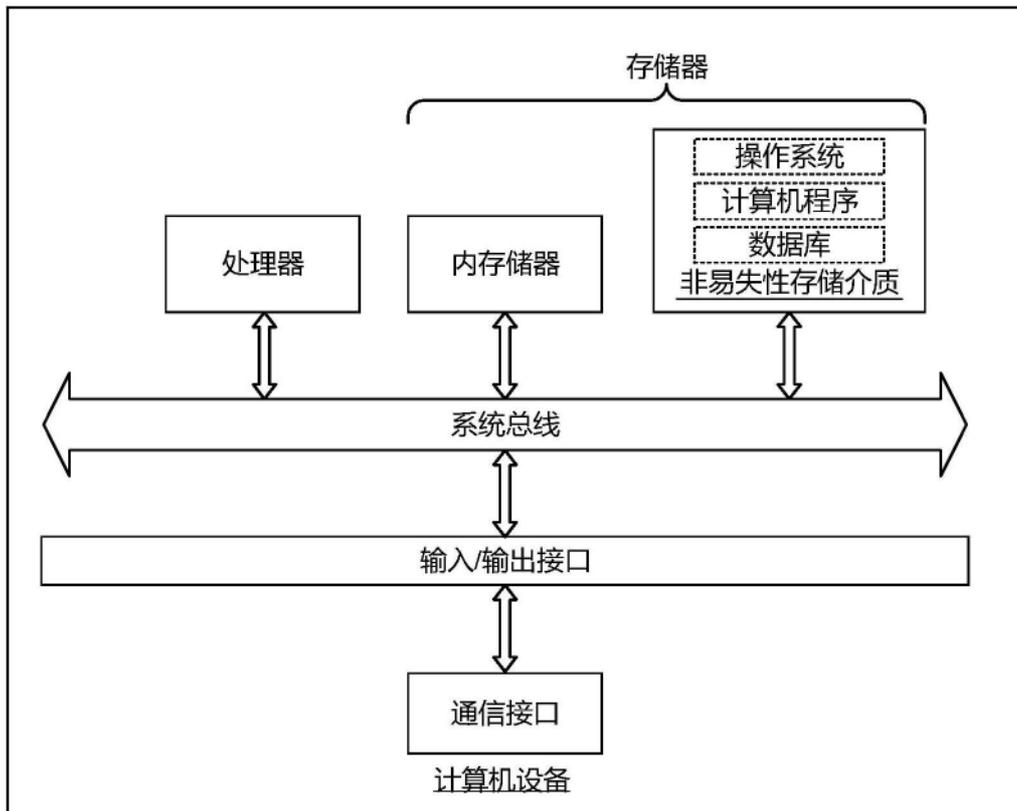


图5

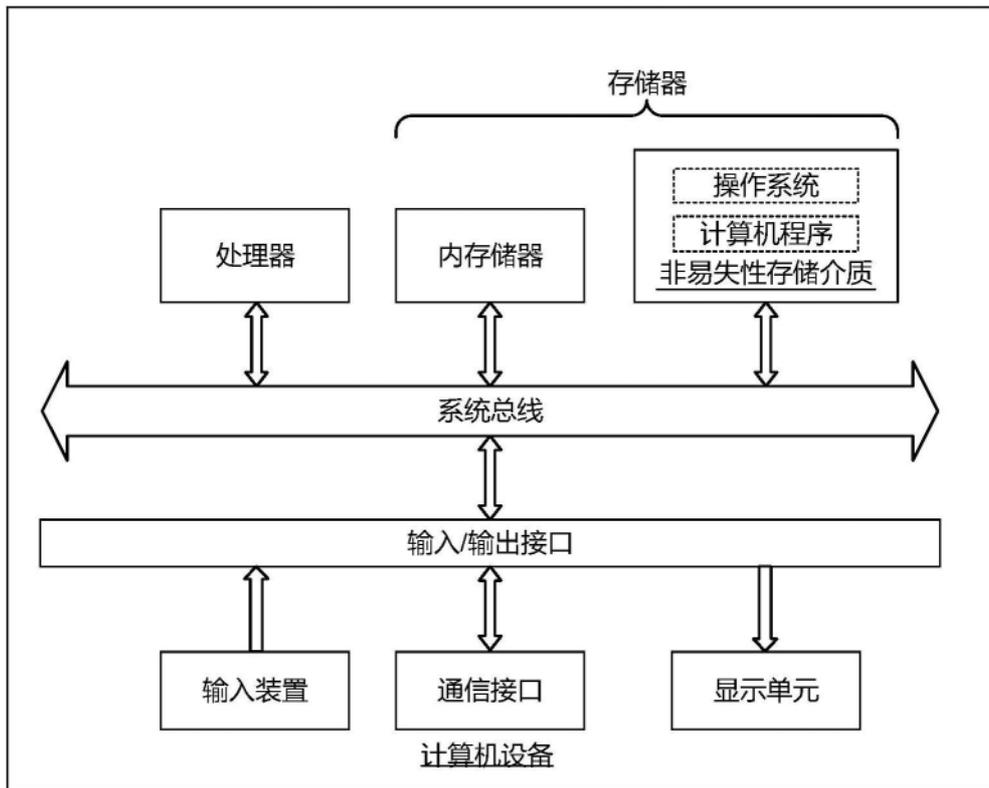


图6