



(12) 发明专利

(10) 授权公告号 CN 115987813 B

(45) 授权公告日 2023.05.30

(21) 申请号 202310266346.5

H04L 67/568 (2022.01)

(22) 申请日 2023.03.13

H04L 67/567 (2022.01)

(65) 同一申请的已公布的文献号

H04L 61/3015 (2022.01)

申请公布号 CN 115987813 A

H04L 67/12 (2022.01)

(43) 申请公布日 2023.04.18

(56) 对比文件

(73) 专利权人 北京集度科技有限公司

CN 101471965 A, 2009.07.01

地址 100176 北京市大兴区北京经济技术

CN 111935327 A, 2020.11.13

开发区宏达北路10号1号楼8层844室

CN 115617515 A, 2023.01.17

(72) 发明人 李睿华

CN 1859172 A, 2006.11.08

WO 2006073399 A1, 2006.07.13

(74) 专利代理机构 北京同达信恒知识产权代理

审查员 高雁

有限公司 11291

专利代理师 黄邃

(51) Int. Cl.

H04L 41/14 (2022.01)

H04L 41/50 (2022.01)

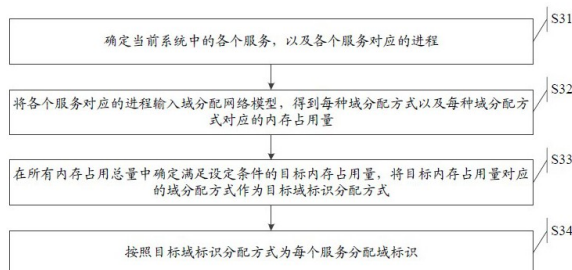
权利要求书2页 说明书10页 附图1页

(54) 发明名称

一种域分配装置、方法、系统及智能车辆

(57) 摘要

本申请公开了一种域分配装置、方法、系统及智能车辆，涉及计算机技术领域，通过该装置将当前系统中的各个服务对应的进程导入到域分配网络模型，然后再计算出输出矩阵，并且在该输出矩阵中包含了域分配方式对应的内存占用量，选择出满足设定条件的目标内存占用量，最终按照目标内存占用量的域标识分配方式为各个服务分配域标识，从而解决当前的域分配方式显著增加了内存消耗的问题，进而减少域分配方式的内存占用量，降低内存消耗。



1. 一种域分配装置,其特征在于,包括处理器和存储器,所述存储器用于存储所述处理器可执行的程序,所述处理器用于读取所述存储器中的程序并执行如下步骤:

确定当前系统中的各个服务,以及各个服务对应的进程,其中,所述服务为所述当前系统中的通信节点提供或者是通信节点使用的服务;

将各个服务对应的进程输入域分配网络模型,得到每种域分配方式以及每种域分配方式对应的内存占用量,其中,每种域分配方式中包含了域标识以及域标识关联的所有服务;

在所有内存占用量中确定满足设定条件的目标内存占用量,将所述目标内存占用量对应的域分配方式作为目标域标识分配方式;

按照所述目标域标识分配方式为每个服务分配域标识。

2. 如权利要求1所述的装置,其特征在于,所述处理器具体被配置为执行:

获取N个服务组合,并将每个服务组合关联的进程构建为进程矩阵,得到N个进程矩阵,其中,所述服务组合中包含至少一个服务,N为大于2的整数;

构建指定网络模型的输出矩阵,其中,所述输出矩阵包含了域标识以及域标识关联的所有服务;

通过所述N个进程矩阵以及所述输出矩阵训练指定网络模型,得到所述域分配网络模型。

3. 如权利要求2所述的装置,其特征在于,所述处理器具体被配置为执行:

将所述当前系统中的所有服务进行N次随机组合,得到由每次随机组合生成的服务组合组成的N个服务组合,其中,所述服务组合中都包含了至少一个服务;

将所述服务组合关联的进程作为矩阵元素添加至矩阵中,得到N个进程矩阵。

4. 如权利要求2所述的装置,其特征在于,所述处理器具体被配置为执行:

确定所述N个服务组合中每个服务组合对应的域标识以及内存占用量;

构建包含内存占用量、域标识以及服务标识的所述指定网络模型的输出矩阵。

5. 如权利要求1所述的装置,其特征在于,所述处理器具体被配置为执行:

在所有内存占用量中确定出最小内存占用量,并将所述最小内存占用量作为所述目标内存占用量。

6. 一种域分配方法,其特征在于,所述方法包括:

确定当前系统中的各个服务,以及各个服务对应的进程,其中,所述服务为所述当前系统中的通信节点提供或者是通信节点使用的服务;

将各个服务对应的进程输入域分配网络模型,得到每种域分配方式以及每种域分配方式对应的内存占用量,其中,每种域分配方式中包含了域标识以及域标识关联的所有服务;

在所有内存占用量中确定满足设定条件的目标内存占用量,将所述目标内存占用量对应的域分配方式作为目标域标识分配方式;

按照所述目标域标识分配方式为每个服务分配域标识。

7. 如权利要求6所述的方法,其特征在于,在确定当前系统中的各个服务,以及各个服务对应的进程之前,所述方法还包括:

获取N个服务组合,并将每个服务组合关联的进程构建为进程矩阵,得到N个进程矩阵,其中,所述服务组合中包含至少一个服务,N为大于2的整数;

构建指定网络模型的输出矩阵,其中,所述输出矩阵包含了域标识以及域标识关联的

所有服务；

通过所述N个进程矩阵以及所述输出矩阵训练指定网络模型，得到所述域分配网络模型。

8. 一种智能车辆，其特征在于，所述智能车辆包括如权利要求1-5中任一项所述的装置。

9. 一种域分配系统，其特征在于，至少包括控制器；所述控制器具体被配置为执行如权利要求6-7中任一项所述的方法。

10. 一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，其特征在于，该计算机程序被处理器执行时，实现权利要求6-7中的任一项所述的方法。

一种域分配装置、方法、系统及智能车辆

技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种域分配装置、方法、系统及智能车辆。

背景技术

[0002] 当前, DDS(Data Distribution Service,数据分发服务)是由OMG(Object Management Group,对象管理组织)定义,基于DCPS(Data-Centric Publish-Subscribe,以数据为中心的发布订阅)模型的一种中间件协议和API标准,因此DDS将系统的组件集成在一起,提供业务和任务关键型物联网(IoT)应用程序所需的低延迟数据连接、极高的可靠性和可扩展架构。

[0003] 根据DDS的应用特点,需要给每个通信节点指定一个域Domain,通信节点只有在相同的Domain内才可以通信,每个通信节点都可以指定多个Domain,根据DDS的特性,通信节点指定一个Domain就会产生一份用于维护该Domain下自身与其他通信节点通信所需的域参与者缓冲区(participant buffer),即需要预分配内存消耗,这种预分配内存消耗的大小与所在Domain内通信节点的数量成正比,例如一个Domain中有5个Node,那么每个Node都要维护4份participant buffer。

[0004] 目前在汽车领域内也使用了DDS作为通信中间件的架构,基本采用一个Domain来管理所有的通信节点,这样保证任意两个通信节点之间都可以通信,但是这种方式也导致每个通信节点都需要保存和域内其他所有通信节点的participant buffer,这样显著增加了内存消耗。

发明内容

[0005] 本发申请提供了一种域分配装置、方法、系统及智能车辆,用以降低DDS中通信节点的内存消耗。

[0006] 第一方面,本申请提供了一种域分配装置,包括处理器和存储器,所述存储器用于存储所述处理器可执行的程序,所述处理器用于读取所述存储器中的程序并执行如下步骤:

[0007] 确定当前系统中的各个服务,以及各个服务对应的进程,其中,所述服务为所述当前系统中的通信节点提供或者是通信节点使用的服务;

[0008] 将各个服务对应的进程输入域分配网络模型,得到每种域分配方式以及每种域分配方式对应的内存占用量,其中,每种域分配方式中包含了域标识以及域标识关联的所有服务;

[0009] 在所有内存占用量中确定满足设定条件的目标内存占用量,将所述目标内存占用量对应的域分配方式作为目标域标识分配方式;

[0010] 按照所述目标域标识分配方式为每个服务分配域标识。

[0011] 通过上述的装置,将当前系统中的各个服务对应的进程导入到域分配网络模型,然后再计算出输出矩阵,并且在输出矩阵中包含了域分配方式对应的内存占用量,选择

出满足设定条件的目标内存占用量,最终按照目标内存占用量的域标识分配方式为各个服务分配域标识,进而保证当前的域标识分配方式可以减少内存的占用量,降低内存消耗。

[0012] 在一种可选的实施例中,所述处理器具体被配置为执行:

[0013] 获取N个服务组合,并将每个服务组合关联的进程构建为进程矩阵,得到N个进程矩阵,其中,所述服务组合中包含至少一个服务,N为大于2的整数;

[0014] 构建指定网络模型的输出矩阵,其中,所述输出矩阵包含了域标识以及域标识关联的所有服务;

[0015] 通过所述N个进程矩阵以及所述输出矩阵训练指定网络模型,得到所述域分配网络模型。

[0016] 在一种可选的实施例中,所述处理器具体被配置为执行:

[0017] 将所述当前系统中的所有服务进行N次随机组合,得到由每次随机组合生成的服务组合组成的N个服务组合,其中,所述服务组合中都包含了至少一个服务;

[0018] 将所述服务组合关联的进程作为矩阵元素添加至矩阵中,得到N个进程矩阵。

[0019] 在一种可选的实施例中,所述处理器具体被配置为执行:

[0020] 确定所述N个服务组合中每个服务组合对应的域标识以及内存占用量;

[0021] 构建包含内存占用量、域标识以及服务标识的所述指定网络模型的输出矩阵。

[0022] 在一种可选的实施例中,所述处理器具体被配置为执行:

[0023] 在所有内存占用量中确定出最小内存占用量,并将所述最小内存占用量作为所述目标内存占用量。

[0024] 第二方面,本申请提供了一种域分配方法,所述方法包括:

[0025] 确定当前系统中的各个服务,以及各个服务对应的进程,其中,所述服务为所述当前系统中的通信节点提供或者是通信节点使用的服务;

[0026] 将各个服务对应的进程输入域分配网络模型,得到每种域分配方式以及每种域分配方式对应的内存占用量,其中,每种域分配方式中包含了域标识以及域标识关联的所有服务;

[0027] 在所有内存占用量中确定满足设定条件的目标内存占用量,将所述目标内存占用量对应的域分配方式作为目标域标识分配方式;

[0028] 按照所述目标域标识分配方式为每个服务分配域标识。

[0029] 通过上述的方法,将当前系统中的各个服务对应的进程导入到域分配网络模型,计算出输出矩阵,并且在该输出矩阵中包含了域分配方式对应的内存占用量,选择出满足设定条件的目标内存占用量,最终按照目标内存占用量的域标识分配方式为各个服务分配域标识,进而保证当前的域标识分配方式可以减少内存的占用量,降低内存消耗。

[0030] 在一种可选的实施例中,在确定当前系统中的各个服务,以及各个服务对应的进程之前,所述方法还包括:

[0031] 获取N个服务组合,并将每个服务组合关联的进程构建为进程矩阵,得到N个进程矩阵,其中,所述服务组合中包含至少一个服务,N为大于2的整数;

[0032] 构建指定网络模型的输出矩阵,其中,所述输出矩阵包含了域标识以及域标识关联的所有服务;

[0033] 通过所述N个进程矩阵以及所述输出矩阵训练指定网络模型,得到所述域分配网

络模型。

[0034] 第三方面,本申请提供了一种智能车辆,所述智能车辆包括上述的任一域分配装置。

[0035] 第四方面,本申请提供了一种域分配系统,至少包括控制器;所述控制器具体被配置为执行上述的任一域分配方法。

[0036] 第五方面,本申请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上存储有计算机程序,该计算机程序被处理器执行时,实现上述的任一域分配方法。

附图说明

[0037] 图1为本申请提供的一种域分配装置的结构示意图;

[0038] 图2为本申请提供的系统中服务与通信节点之间关系示意图;

[0039] 图3为本申请提供的一种域分配方法的流程图。

具体实施方式

[0040] 为了使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明作进一步地详细描述,显然,所描述的实施例仅仅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前提下所获得的所有其它实施例,都属于本发明保护的范围。

[0041] 本发明实施例中术语“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。

[0042] 本发明实施例描述的应用场景是为了更加清楚的说明本发明实施例的技术方案,并不构成对于本发明实施例提供的技术方案的限定,本领域普通技术人员可知,随着新应用场景的出现,本发明实施例提供的技术方案对于类似的技术问题,同样适用。其中,在本发明的描述中,除非另有说明,“多个”的含义是两个或两个以上。

[0043] 当前, DDS是由OMG定义,基于DCPS模型的一种中间件协议和API标准,因此DDS将系统的组件集成在一起,提供业务和任务关键型物联网 (IoT) 应用程序所需的低延迟数据连接、极高的可靠性和可扩展架构。

[0044] 根据DDS的应用特点,需要给每个通信节点指定一个域Domain,通信节点只有在相同的Domain内才可以通信,每个通信节点都可以指定多个Domain,根据DDS的特性,通信节点指定一个Domain就会产生一份用于维护该Domain下自身与其他通信节点通信所需的域参与者缓冲区(participant buffer),即需要预分配内存消耗,这种预分配内存消耗的大小与所在Domain内通信节点的数量成正比,例如一个Domain中有5个Node,那么每个Node都要维护4份participant buffer。

[0045] 目前在汽车领域内也使用了DDS作为通信中间件的架构,基本采用一个Domain来管理所有的通信节点,这样保证任意两个通信节点之间都可以通信,但是这种方式也导致每个通信节点都需要保存和域内其他所有通信节点的participant buffer,这样显著增加了内存消耗。

[0046] 为了解决上述问题,本申请实施例中提供了一种域分配装置,通过该装置将当前

系统中的各个服务按照不同的方式分配域标识,从而得到各种不同的域标识分配方式,然后再计算出每种域分配方式对应的内存占用量,在内存占用量中确定出满足设定条件的目标内存占用量,最终按照目标内存占用量的域标识分配方式为各个服务分配域标识,进而保证当前的域标识分配方式可以减少内存的占用量,降低内存消耗。

[0047] 参照图1所示为本申请实施例提供的一种域分配装置,该装置包括处理器10以及存储器11,存储器11用于存储处理器10可执行的程序,处理器10用于读取存储器11中的程序并执行如下步骤:

[0048] 确定当前系统中的各个服务,以及各个服务对应的进程;

[0049] 参照图2所示,该服务为当前系统中的通信节点提供或者是通信节点使用的服务,在图2中还包含了各个服务与通信节点之间关联;

[0050] 将各个服务对应的进程输入域分配网络模型,得到每种域分配方式以及每种域分配方式对应的内存占用量,其中,每种域分配方式中包含了域标识以及域标识关联的所有服务;

[0051] 在所有内存占用量中确定满足设定条件的目标内存占用量,将目标内存占用量对应的域分配方式作为目标域标识分配方式;

[0052] 按照目标域标识分配方式为每个服务分配域标识。

[0053] 具体来讲,在确定当前系统中的各个服务之前,首先需要对指定网络模型进行训练来得到该域分配网络模型,具体实现方法如下:

[0054] 首先在当前系统中确定出所有服务,然后将这些服务进行随机组合,比如说,当前系统中存在服务1、服务2、服务3、服务4,将服务1与服务2进行组合、服务3与服务4进行组合,服务1与服务3进行组合,得到多个服务组合,从而得到由每次随机组合生成的服务组合组成的N个服务组合。在每个服务组合中都包含至少一个服务。

[0055] 这里需要说明是,单独一个服务也作为一个组合,按照上述的举例来说,服务1、服务2、服务3、服务4都是一个服务组合。

[0056] 在得到N个服务组合之后,将服务组合关联的进程作为矩阵元素添加到矩阵中,得到N个进程矩阵。

[0057] 举例来说,服务1关联的进程为p00,p01,p02...p0m,服务2关联的进程为p10,p11,p12...p1m,服务3关联的进程为p20,p21,p22...p2m,服务4关联的进程为p30,p31,p32...p3m。

[0058] 此时由服务1、服务2、服务3、服务4单独可以作为一个服务组合,则由服务1单独作为一个服务组合对应的进程矩阵为:

[0059] [p00 p01 p0m]

[0060] [0 0 0]

[0061] ...

[0062] [0 0 0]

[0063] 这里需要说明的是,该进程矩阵的行数与当前系统中的所有服务的数量有关,也就是当前系统中的服务为4个,则进程矩阵中包含4行,每一行代表一个服务所关联的进程。

[0064] 在上述的举例中,由于是服务1单独作为一个服务组合,因此进程矩阵中的第一行为服务1所关联的进程,其他行则用“0”进行填充。当然其他服务单独构建进程矩阵的方式与服务1构建进程矩阵的方式相同,此处就不再赘述。

[0065] 当然,除了由单个服务构建进程矩阵之外,还可以是多个服务的组合构建进程矩阵,比如服务1与服务3组合构建进程矩阵,具体如下:

[0066] [p00 p01 p0m]

[0067] [p20 p21 p2m]

[0068] ...

[0069] [0 0 0]

[0070] 当然,还可以是由服务1、服务2、服务3、服务4共同构建进程矩阵,具体如下:

[0071] [p00 p01 p0m]

[0072] [p10 p11 p1m]

[0073] [P20 p21 p2m]

[0074] [p30 p31 p3m]

[0075] 按照上述的方法就可以将N个服务组合分别构建出对应的进程矩阵,从而得到N个进程矩阵。这N个进程矩阵作为训练指定网络模型的输入。

[0076] 除了输入之外,训练该指定网络模型还需要对应的输出,该指定网络模型的输出为内存占用量、域标识以及服务标识构建的输出矩阵。具体来讲,首先确定N个服务组合中每个服务组合对应的域标识以及内存占用量,构建包含内存占用量、域标识以及服务标识的指定网络模型的输出矩阵。

[0077] 举例来说,若是当前系统中总共16个进程,此时输出矩阵中一行最多16位,如下为服务1、服务2、服务3、服务4单独构建进程矩阵对应的输出矩阵:

[0078] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0]

[0079] [id1 s1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0080] [id2 s2 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0081] [id3 s3 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0082] [id4 s4 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0083] 其中,300表征内存占用量,id1、id2、id3、id4分别表征不同的域标识,s1、s2、s3、s4分别表征的服务1、服务2、服务3、服务4的标识。

[0084] 同理,若是服务1、服务2、服务3组成一个服务组合,服务4单独组成一个服务组合,对应的输出矩阵为:

[0085] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0]

[0086] [id4 s1 s2 s3 0 0 0 0 0 0 0 0 0 0 0 0]

[0087] [id5 s4 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0088] 其中,200表征内存占用量,id4、id5分别表征不同的域标识,s1、s2、s3、s4分别表征的服务1、服务2、服务3、服务4的标识。

[0089] 另外,若是服务1、服务2组成一个服务组合,服务3和服务4组成一个服务组合,则对应的输出矩阵为:

[0090] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]

[0091] [id6 s1 s2 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0092] [id7 s3 s4 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0093] 其中,100表征内存占用量,id6、id6分别表征不同的域标识,s1、s2、s3、s4分别表

征的服务1、服务2、服务3、服务4的标识。

[0094] 当然,上述举例只是说明了其中几种服务组合对应的输出矩阵,其他可能的服务组合就不再一一举例进行说明。

[0095] 在确定好指定网络模型的输入矩阵以及输出矩阵之后,将通过输入矩阵以及输出矩阵对指定网络模型进行训练,在训练过程中需要通过指定网络模型的输出矩阵进一步的确定是否对指定网络模型的训练参数进行调整。

[0096] 具体来讲,在指定网络模型输出对应的输出矩阵之后,计算出输出矩阵与实际输出矩阵之间的差异值,并通过该差异值来调整指定网络模型的训练参数,直至指定网络模型的输出矩阵与实际输出矩阵之间的差异值在设定范围内,此时将指定网络模型作为域分配网络模型。这里需要说明是,该实际输出矩阵是预先设定的并且是输入矩阵对应的输出矩阵。

[0097] 通过上述的方法,能够准确的得到域分配网络模型,从而通过该域分配网络模型能够确定出当前系统中各个服务的域分配方式。

[0098] 因此,在得到上述的域分配网络模型之后,确定当前系统中的各个服务,以及各个服务对应的进程。将各个服务对应的进程构建输入矩阵,将该输入矩阵导入到域分配网络模型中。该域分配网络模型将直接计算出输出矩阵,此处的输出矩阵不止一个,每一个输出矩阵都表征了一种域分配方式。

[0099] 比如说,当前系统中包含了6个服务(服务1、服务2、服务3、服务4、服务5、服务6),这6个服务包含了6*m个进程。此时对应的输入矩阵为:

[0100] [p10 p11 p1m]

[0101] [p20 p21 p2m]

[0102] [P30 p31 p3m]

[0103] [p40 p41 p4m]

[0104] [p50 p51 p5m]

[0105] [p60 p61 p6m]

[0106] 将该输入矩阵导入的到域分配网络模型之后,该域分配网络模型将输出各种输出矩阵,比如输出矩阵1、输出矩阵2、输出矩阵3、输出矩阵4。

[0107] 其中,输出矩阵1为:

[0108] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0]

[0109] [id1 s1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0110] [id2 s2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0111] [id3 s3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0112] [id4 s4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0113] [id5 s5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0114] [id6 s6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0115] 这里输出矩阵表征的是,服务1、服务2、服务3、服务4、服务5、服务6都单独分配了一个域标识,这种域分配方式对应的内存占用量为500。

[0116] 其中,输出矩阵2为:

[0117] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0]

[0118] [id1 s1 s2 s3 0 0 0 0 0 0 0 0 0 0 0 0]

[0119] [id2 s4 s5 0 0 0 0 0 0 0 0 0 0 0 0]

[0120] [id3 s6 0 0 0 0 0 0 0 0 0 0 0 0]

[0121] 这里输出矩阵表征的是,服务1、服务2、服务3为分配了同一个域标识,服务4和服务5分配了同一个域标识,服务6单独分配了一个域标识,这种域分配方式对应的内存占用量为400。

[0122] 输出矩阵3为:

[0123] [0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0]

[0124] [id1 s1 s2 0 0 0 0 0 0 0 0 0 0 0 0]

[0125] [id2 s3 s4 0 0 0 0 0 0 0 0 0 0 0 0]

[0126] [id3 s5 s6 0 0 0 0 0 0 0 0 0 0 0 0]

[0127] 这里输出矩阵表征的是,服务1、服务2为分配了同一个域标识,服务3和服务4分配了同一个域标识,服务5和服务6分配了同一个域标识,这种域分配方式对应的内存占用量为300。

[0128] 输出矩阵4为:

[0129] [0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0]

[0130] [id1 s1 s2 s3 0 0 0 0 0 0 0 0 0 0 0 0]

[0131] [id2 s4 s5 s6 0 0 0 0 0 0 0 0 0 0 0 0]

[0132] 这里输出矩阵表征的是,服务1、服务2、服务3分配了同一个域标识,服务4、服务5、服务6分配了同一个域标识,这种域分配方式对应的内存占用量为200。

[0133] 在得到上述的各种输出矩阵之后,在所有的输出矩阵中筛选出内存占用量满足设定条件的目标内存占用量,比如说按照内存由小到大的方式进行排序,将前2个内存占用量中的任一个内存占用量作为目标内存占用量,比如上述举例中,将内存占用量为200、内存占用量为300的作为目标内存占用量,将该输出矩阵对应的域分配方式作为目标域分配方式。

[0134] 比如说,在上述举例中,确定内存占用量为300为目标内存占用量,此时将该输出矩阵对应的域分配方式作为目标域分配的方式,也就是将服务1、服务2为分配同一个域标识,服务3和服务4分配同一个域标识,服务5和服务6分配同一个域标识。

[0135] 当然,在实际应用场景中可以将内存占用量最小的域分配方式作为目标域分配方式。比如说将内存占用量为200为目标内存占用量,此时服务1、服务2、服务3分配同一个域标识,服务4、服务5、服务6分配同一个域标识。

[0136] 通过上述方式可以避免所有服务使用同一个域造成总的内存占用量较大的问题,并且通过上述的方法可以最大程度上的减少域分配不合理导致的内存消耗。

[0137] 基于相同的发明构思,本发明实施例还提供了一种域分配方法,该方法解决问题的原理与上述任一的域分配装置相似,因此该方法的实施可参见装置的实施,重复之处不再赘述。

[0138] 参照图3所示,该域分配方法的具有实施流程包括:

[0139] S31,确定当前系统中的各个服务,以及各个服务对应的进程;

[0140] 这里需要说明是,服务为当前系统中的通信节点提供或者是通信节点使用的服

务；

[0141] S32,将各个服务对应的进程输入域分配网络模型,得到每种域分配方式以及每种域分配方式对应的内存占用量；

[0142] 这里需要说明是,每种域分配方式中包含了域标识以及域标识关联的所有服务；

[0143] S33,在所有内存占用量中确定满足设定条件的目标内存占用量,将目标内存占用量对应的域分配方式作为目标域标识分配方式；

[0144] S34,按照目标域标识分配方式为每个服务分配域标识。

[0145] 在一种可选的实施例中,获取N个服务组合,并将每个服务组合关联的进程构建为进程矩阵,得到N个进程矩阵,构建指定网络模型的输出矩阵,通过N个进程矩阵以及输出矩阵训练指定网络模型,得到域分配网络模型。

[0146] 在一种可选的实施例中,将所述当前系统中的所有服务进行N次随机组合,得到由每次随机组合生成的服务组合组成的N个服务组合,其中,所述服务组合中都包含了至少一个服务；

[0147] 将所述服务组合关联的进程作为矩阵元素添加至矩阵中,得到N个进程矩阵。

[0148] 在一种可选的实施例中,确定所述N个服务组合中每个服务组合对应的域标识以及内存占用量；

[0149] 构建包含内存占用量、域标识以及服务标识的所述指定网络模型的输出矩阵。

[0150] 具体如何得到域分配网络模型以及如何确定出对应的域分配方式在前述的实施例中已经详细说明,此处就不再赘述。

[0151] 通过上述的方法,将当前系统中的各个服务对应的进程导入到域分配网络模型,然后再计算出输出矩阵,并且在输出矩阵中包含了域分配方式对应的内存占用量,选择出满足设定条件的目标内存占用量,最终按照目标内存占用量的域标识分配方式为各个服务分配域标识,进而保证当前的域标识分配方式可以减少内存的占用量,降低内存消耗。

[0152] 基于相同的发明构思,本发明实施例还提供了一种域分配系统,该系统解决问题的原理与上述任一域分配方法相似,因此该系统的实施可参见方法的实施,重复之处不再赘述。

[0153] 本申请实施例中所提供的系统至少包括控制器；所述控制器具体被配置为执行如下步骤：

[0154] 确定当前系统中的各个服务,以及各个服务对应的进程,其中,所述服务为所述当前系统中的通信节点提供或者是通信节点使用的服务；

[0155] 将各个服务对应的进程输入域分配网络模型,得到每种域分配方式以及每种域分配方式对应的内存占用量,其中,每种域分配方式中包含了域标识以及域标识关联的所有服务；

[0156] 在所有内存占用量中确定满足设定条件的目标内存占用量,将所述目标内存占用量对应的域分配方式作为目标域标识分配方式；

[0157] 按照所述目标域标识分配方式为每个服务分配域标识。

[0158] 在一种可选的实施例中,所述控制器具体被配置为执行：

[0159] 获取N个服务组合,并将每个服务组合关联的进程构建为进程矩阵,得到N个进程矩阵,其中,所述服务组合中包含至少一个服务,N为大于2的整数；

[0160] 构建指定网络模型的输出矩阵,其中,所述输出矩阵包含了域标识以及域标识关联的所有服务;

[0161] 通过所述N个进程矩阵以及所述输出矩阵训练指定网络模型,得到所述域分配网络模型。

[0162] 在一种可选的实施例中,所述控制器具体被配置为执行:

[0163] 将所述当前系统中的所有服务进行N次随机组合,得到由每次随机组合生成的服务组合组成的N个服务组合,其中,所述服务组合中都包含了至少一个服务;

[0164] 将所述服务组合关联的进程作为矩阵元素添加至矩阵中,得到N个进程矩阵。

[0165] 在一种可选的实施例中,所述控制器具体被配置为执行:

[0166] 确定所述N个服务组合中每个服务组合对应的域标识以及内存占用量;

[0167] 构建包含内存占用量、域标识以及服务标识的所述指定网络模型的输出矩阵。

[0168] 基于相同的发明构思,本发明实施例还提供了一种智能车辆,该智能车辆包括上述任一的域分配装置。该智能车辆用于基于任一的域分配装置,实现对系统中服务进行域标识分配的功能。

[0169] 基于同一发明构思,本公开实施例提供一种计算机存储介质,计算机存储介质包括:计算机程序代码,当计算机程序代码在计算机上运行时,使得计算机执行如前文论述任一的域分配方法。由于上述计算机存储介质解决问题的原理与域分配方法相似,因此上述计算机存储介质的实施可以参见方法的实施,重复之处不再赘述。

[0170] 在具体的实施过程中,计算机存储介质可以包括:通用串行总线闪存盘(USB, Universal Serial Bus Flash Drive)、移动硬盘、只读存储器(ROM, Read-Only Memory)、随机存取存储器(RAM, Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的存储介质。

[0171] 基于同一发明构思,本公开实施例还提供了一种计算机程序产品,该计算机程序产品包括:计算机程序代码,当该计算机程序代码在计算机上运行时,使得计算机执行如前文论述任一的域分配方法。由于上述计算机程序产品解决问题的原理与域分配方法相似,因此上述计算机程序产品的实施可以参见方法的实施,重复之处不再赘述。

[0172] 计算机程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以是但不限于电、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。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0173] 本申请中的方法可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机程序或指令。在计算机上加载和执行所述计算机程序或指令时,全部或部分地执行本申请所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、网络设备、用户设备、核心网设备、OAM或者其它可编程装置。

[0174] 所述计算机程序或指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一个计算机可读存储介质传输,例如,所述计算机程序或指令可以从一个

网站站点、计算机、服务器或数据中心通过有线或无线方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是集成一个或多个可用介质的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质,例如,软盘、硬盘、磁带;也可以是光介质,例如,数字视频光盘;还可以是半导体介质,例如,固态硬盘。该计算机可读存储介质可以是易失性或非易失性存储介质,或可包括易失性和非易失性两种类型的存储介质。

[0175] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器和光学存储器等)上实施的计算机程序产品的形式。

[0176] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的设备。

[0177] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令设备的制品,该指令设备实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0178] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0179] 显然,本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样,倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内,则本发明也意图包含这些改动和变型在内。



图 1

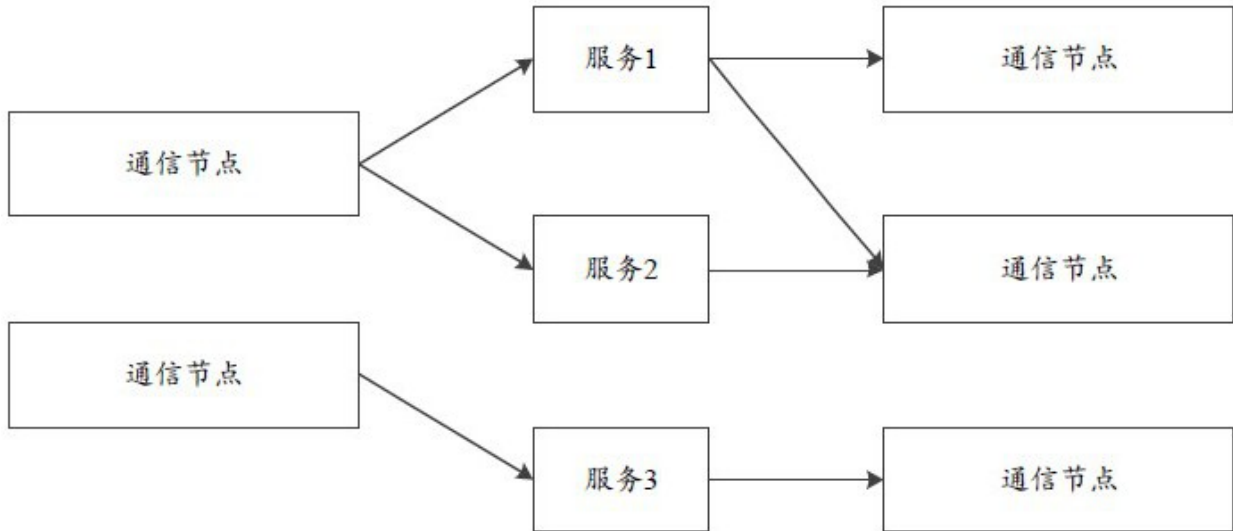


图 2

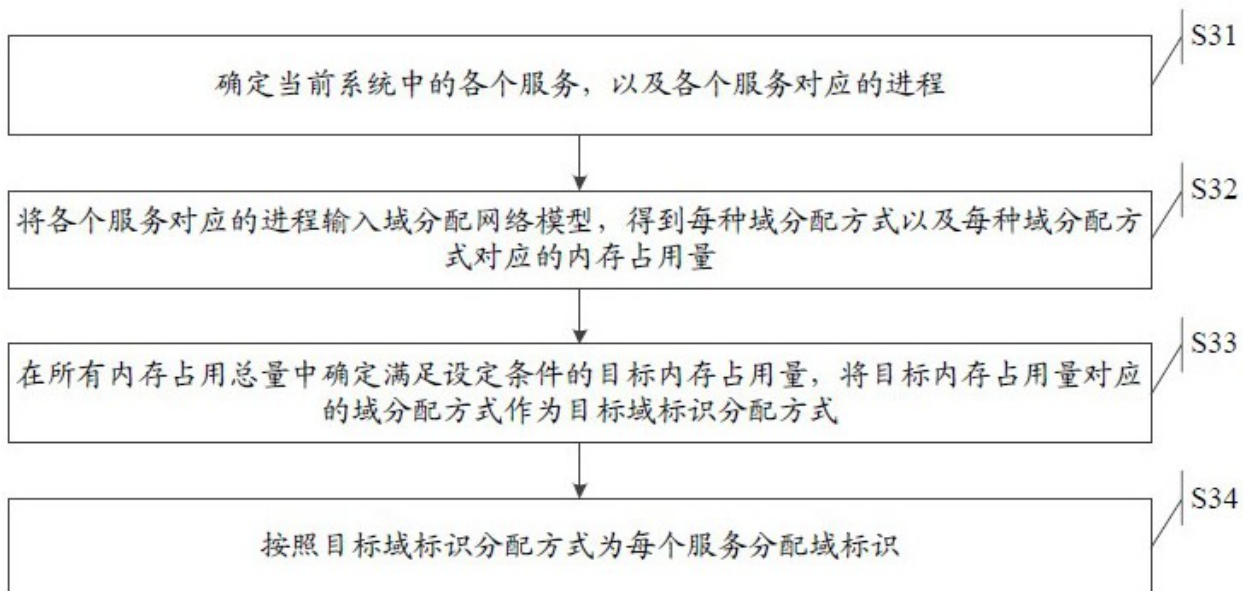


图 3