



(12) 发明专利申请

(10) 申请公布号 CN 113687962 A

(43) 申请公布日 2021. 11. 23

(21) 申请号 202110962576.6

(22) 申请日 2021.08.20

(71) 申请人 济南浪潮数据技术有限公司

地址 250101 山东省济南市自由贸易试验区  
济南片区浪潮路1036号浪潮科技园  
S05楼S311室

(72) 发明人 朱文龙

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

代理人 纪志超

(51) Int. Cl.

G06F 9/54 (2006.01)

G06F 9/48 (2006.01)

G06F 9/50 (2006.01)

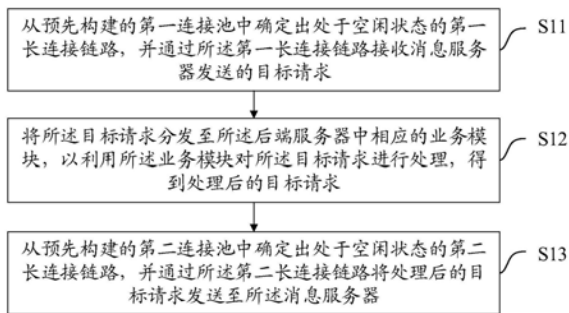
权利要求书2页 说明书7页 附图4页

(54) 发明名称

一种请求处理方法、装置、设备及存储介质

(57) 摘要

本申请公开了一种请求处理方法、装置、设备及存储介质,应用于后端服务器,包括:从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过第一长连接链路接收消息服务器发送的目标请求;将目标请求分发至后端服务器中相应的业务模块,以利用业务模块对目标请求进行处理,得到处理后的目标请求;从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过第二长连接链路将处理后的目标请求发送至消息服务器。本申请通过第一连接池中的第一长连接链路接收消息服务器发送的目标请求,并通过第二连接池中的第二长连接链路返回处理后的目标请求,基于预先构建的连接池复用长连接链路,降低消息服务器压力,提高请求处理效率。



1. 一种请求处理方法,其特征在于,应用于后端服务器,包括:

从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;

将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;

从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器;

其中,所述将所述目标请求分发至所述后端服务器中相应的业务模块,包括:

如果所述目标请求不止一个,则获取各个所述目标请求对应的优先级,并按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得到对应的请求序列;

将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块。

2. 根据权利要求1所述的请求处理方法,其特征在于,所述通过所述第一长连接链路接收消息服务器发送的目标请求之前,还包括:

请求端通过WEB页面或者命令行界面将目标消息发送至消息服务器;

相应的,所述通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器之后,还包括:

所述消息服务器将处理后的目标请求返回至所述请求端。

3. 根据权利要求1所述的请求处理方法,其特征在于,所述按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得到对应的请求序列之后,还包括:

通过消息队列的方式对所述请求序列中的各个所述目标请求进行存储。

4. 根据权利要求1所述的请求处理方法,其特征在于,所述将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块,包括:

确定各个所述目标请求对应的标识码,并根据各个所述目标请求对应的所述标识码将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块;其中,所述标识码与所述业务模块之间存在唯一对应关系。

5. 根据权利要求1至4任一项所述的请求处理方法,其特征在于,所述消息服务器为RabbitMQ消息中间件。

6. 根据权利要求5所述的请求处理方法,其特征在于,还包括:

部署单个RabbitMQ消息中间件或RabbitMQ消息中间件集群。

7. 一种请求处理装置,其特征在于,应用于后端服务器,包括:

消息接收模块,用于从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;

消息处理模块,用于将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;

消息发送模块,用于从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器;

其中,所述请求处理装置,还用于如果所述目标请求不止一个,则获取各个所述目标请求对应的优先级,并按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得

到对应的请求序列;以及将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块。

8.一种电子设备,其特征在于,所述电子设备包括处理器和存储器;其中所述存储器用于存储计算机程序,所述计算机程序由所述处理器加载并执行以实现如权利要求1至6任一项所述的请求处理方法。

9.一种计算机可读存储介质,其特征在于,用于存储计算机可执行指令,所述计算机可执行指令被处理器加载并执行时,实现如权利要求1至6任一项所述的请求处理方法。

## 一种请求处理方法、装置、设备及存储介质

### 技术领域

[0001] 本发明涉及计算机技术领域,特别涉及一种请求处理方法、装置、设备及存储介质。

### 背景技术

[0002] 在现有存储管理软件的架构中根据后端不同功能拆分的模块,集群所有节点的每个模块都保持和单个消息服务节点的一个消息服务长连接,这样一来单个消息服务节点的压力很大。但是除了性能和缓存等定时执行收集信息的模块外,其他业务模块基本都处于不活跃状态,当没有CLI和前端页面操作的请求命令时模块连接处于“空跑”状态,浪费连接资源。同时各个模块处理请求之后又各自临时申请创建新的连接用于发送处理结果,每次接收一个处理请求就新建一次连接,资源开销很大,当节点和模块数很多的情况下对该单个消息服务节点造成很大的压力,降低请求响应效率。

[0003] 因此,如何提高存储管理软件架构的请求处理效率,是本领域亟待解决的技术问题。

### 发明内容

[0004] 有鉴于此,本发明的目的在于提供一种请求处理方法、装置、设备及存储介质,能够基于预先构建的连接池复用长连接链路,降低消息服务器压力,提高请求处理效率。其具体方案如下:

[0005] 本申请的第一方面提供了一种请求处理方法,应用于后端服务器,包括:

[0006] 从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;

[0007] 将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;

[0008] 从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。

[0009] 可选的,所述通过所述第一长连接链路接收消息服务器发送的目标请求之前,还包括:

[0010] 请求端通过WEB页面或者命令行界面将目标消息发送至消息服务器;

[0011] 相应的,所述通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器之后,还包括:

[0012] 所述消息服务器将处理后的目标请求返回至所述请求端。

[0013] 可选的,所述将所述目标请求分发至所述后端服务器中相应的业务模块,包括:

[0014] 如果所述目标请求不止一个,则获取各个所述目标请求对应的优先级,并按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得到对应的请求序列;

[0015] 将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业

务模块。

[0016] 可选的,所述按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得到对应的请求序列之后,还包括:

[0017] 通过消息队列的方式对所述请求序列中的各个所述目标请求进行存储。

[0018] 可选的,所述将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块,包括:

[0019] 确定各个所述目标请求对应的标识码,并根据各个所述目标请求对应的所述标识码将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块;其中,所述标识码与所述业务模块之间存在唯一对应关系。

[0020] 可选的,所述消息服务器为RabbitMQ消息中间件。

[0021] 可选的,所述请求处理方法,还包括:

[0022] 部署单个RabbitMQ消息中间件或RabbitMQ消息中间件集群。

[0023] 本申请的第二方面提供了一种请求处理装置,应用于后端服务器,包括:

[0024] 消息接收模块,用于从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;

[0025] 消息处理模块,用于将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;

[0026] 消息发送模块,用于从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。

[0027] 本申请的第三方面提供了一种电子设备,所述电子设备包括处理器和存储器;其中所述存储器用于存储计算机程序,所述计算机程序由所述处理器加载并执行以实现前述请求处理方法。

[0028] 本申请的第四方面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中存储有计算机可执行指令,所述计算机可执行指令被处理器加载并执行时,实现前述请求处理方法。

[0029] 本申请中,先从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;然后将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;最后从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。本申请通过第一连接池中处于空闲状态的第一长连接链路接收消息服务器发送的目标请求,并通过第二连接池中处于空闲状态的第二长连接链路返回处理后的目标请求,基于预先构建的连接池复用长连接链路,降低消息服务器压力,提高请求处理效率。

## 附图说明

[0030] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的实施例,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,还可以根据提供的附图获得其他的附图。

- [0031] 图1为本申请提供了一种请求处理方法流程图；
- [0032] 图2为本申请提供了一种现有的请求处理方法示意图；
- [0033] 图3为本申请提供了一种现有的单节点RabbitMQ服务与各模块建立长连接方法示意图；
- [0034] 图4为本申请提供了一种具体的请求处理方法流程图；
- [0035] 图5为本申请提供了一种具体的请求处理方法示意图；
- [0036] 图6为本申请提供了一种请求处理装置结构示意图；
- [0037] 图7为本申请提供了一种请求处理电子设备结构图。

### 具体实施方式

[0038] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,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他实施例,都属于本发明保护的范围。

[0039] 在现有存储管理软件的架构中根据后端不同功能拆分的模块,集群所有节点的每个模块都保持和单个消息服务节点的一个消息服务长连接,这样一来单个消息服务节点的压力很大。但是除了性能和缓存等定时执行收集信息的模块外,其他业务模块基本都处于不活跃状态,当没有CLI和前端页面操作的请求命令时模块连接处于“空跑”状态,浪费连接资源。同时各个模块处理请求之后又各自临时申请创建新的连接用于发送处理结果,每次接收一个处理请求就新建一次连接,资源开销很大,当节点和模块数很多的情况下对该单个消息服务节点造成很大的压力,降低请求响应效率。针对上述技术缺陷,本申请提供一种请求处理方案,通过第一连接池中处于空闲状态的第一长连接链路接收消息服务器发送的目标请求,并通过第二连接池中处于空闲状态的第二长连接链路返回处理后的目标请求,基于预先构建的连接池复用长连接链路,降低消息服务器压力,提高请求处理效率。

[0040] 图1为本申请实施例提供了一种请求处理方法流程图。参见图1所示,该请求处理方法应用于后端服务器应用于后端服务器,包括:

[0041] S11:从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求。

[0042] 本实施例中,从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求。不难理解,在所述通过所述第一长连接链路接收消息服务器发送的目标请求之前,请求端通过WEB页面或者命令行界面(CLI,command-line interface)将所述目标请求发送至所述消息服务器。相应的,在对所述目标请求进行处理之后,所述消息服务器还应该将处理后的目标请求返回至所述请求端。本实施例中,所述消息服务器可以为RabbitMQ消息中间件,也即可以部署单个RabbitMQ消息中间件或RabbitMQ消息中间件集群。其中,RabbitMQ Server为实现高级消息队列协议(AMQP)的开源消息中间件。

[0043] 图2为现有的请求处理示意图,各节点中各模块各自独立创建消息长连接到单个RabbitMQ服务器节点处理请求接收和请求发送。其中,单个RabbitMQ服务器节点与各模块与单个RabbitMQ服务节点之间建立长连接的方式具体如图3所示,该情况存在单个

RabbitMQ服务节点接收到的长连接数过多问题和空闲长连接利用效率低的问题。本实施例中,在单节点RabbitMQ对外提供服务的场景下,通过创建一个消息服务长连接池,来统一接收请求消息,降低单RabbitMQ消息长连接的数量。这样复用消息队列长连接,避免了不断的创建和销毁临时连接的开销。其中,所述第一连接池为接收连接池。

[0044] S12:将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求。

[0045] 本实施例中,所述后端服务器接收到所述目标请求后,将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求。具体的,基于所述目标请求与所述业务模块之间的对应的关系来分发所述目标请求,所述对应关系需要预先构建,本申请实施例对此不进行限定。例如,所述对应关系可以通过存在唯一对应关系的标识码实现,对应的,所述后端服务器首先确定各个所述目标请求对应的标识码,然后根据各个所述目标请求对应的所述标识码将所述目标请求依次分发至所述后端服务器中相应的业务模块。不难理解,所述目标请求以数据包的形式传输,请求数据包中包含标识码等相关信息。

[0046] S13:从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。

[0047] 本实施例中,所述后端服务器中的所述业务模块对相应的所述目标请求进行处理后,需要从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器,处理后的所述目标请求经所述消息服务器返回至所述请求端。其中,所述第二连接池可以为发送连接池。通过对单个RabbitMQ服务节点创建接收连接池和发送连接池来统一处理请求的接收和发送,可以根据系统压力控制长连接的数量,复用创建长连接的同时避免频繁的创建和销毁长连接的开销,当集群规模不断增大的情况下可以有效减少节点到单个RabbitMQ的长连接数,降低单个RabbitMQ服务的压力,同时避免长连接创建的流程,加快请求的处理速度。

[0048] 可见,本申请实施例先从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;然后将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;最后从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。本申请实施例通过第一连接池中处于空闲状态的第一长连接链路接收消息服务器发送的目标请求,并通过第二连接池中处于空闲状态的第二长连接链路返回处理后的目标请求,基于预先构建的连接池复用长连接链路,降低消息服务器压力,提高请求处理效率。

[0049] 图4为本申请实施例提供的一种具体的请求处理方法流程图。参见图4所示,该请求处理方法包括:

[0050] S21:从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求。

[0051] 本实施例中,关于上述步骤S21的具体过程,可以参考前述实施例中公开的相应内容,在此不再进行赘述。需要说明的是,所述第一连接池为Consumer连接池,相应的,所述第一长连接链路为与RabbitMQ server的消费者长连接。

[0052] S22:如果所述目标请求不止一个,则获取各个所述目标请求对应的优先级,并按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得到对应的请求序列。

[0053] S23:将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块。

[0054] 本实施例中,在高并发场景下,从RabbitMQ服务器集群中同时转发大量所述目标请求,可根据所述目标请求的优先级进行分发处理,命令行CLI或者前端页面发送请求中带有优先级参数,其中,发布请求的应用程序可以为每个请求分配优先级,具体参照图5。也即先获取各个所述目标请求对应的优先级,然后按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得到对应的请求序列。另外,一般通过消息队列的方式对所述请求序列中的各个所述目标请求进行存储。

[0055] 具体来说,目标请求发送到单个RabbitMQ消息服务节点后根据优先级自动重新排序插入接收消息队列中,Consumer连接池根据所述目标请求的优先级从高到低不断取出所述目标请求。然后,将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块。基于上述实施例,确定各个所述目标请求对应的标识码,并根据各个所述目标请求对应的所述标识码将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块;其中,所述标识码与所述业务模块之间存在唯一对应关系。

[0056] S24:从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。

[0057] 本实施例中,关于上述步骤S24的具体过程,可以参考前述实施例中公开的相应内容,在此不再进行赘述。需要说明的是,所述第二连接池为Connection连接池。

[0058] 可见,本申请实施例一方面在非RabbitMQ集群模式下,通过创建连接池来统一处理请求的接收和发送,当扩容节点规模不断增大和节点中模块数增多的情况下可以有效减少单个RabbitMQ服务节点维护的长连接数,降低RabbitMQ消息服务器的压力,同时由于没有临时消息长连接的建立流程,可以显著加快消息的发送速度。另一方面发布请求的应用程序为每个请求分配优先级,根据优先级进行请求的分发处理,保证了在高并发多类型请求的场景下对重要和关键的请求优先响应处理,有效改善用户体验。

[0059] 参见图6所示,本申请实施例还相应公开了一种请求处理装置,应用于后端服务器,包括:

[0060] 消息接收模块11,用于从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;

[0061] 消息处理模块12,用于将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;

[0062] 消息发送模块13,用于从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。

[0063] 可见,本申请实施例先从预先构建的第一连接池中确定出处于空闲状态的第一长连接链路,并通过所述第一长连接链路接收消息服务器发送的目标请求;然后将所述目标请求分发至所述后端服务器中相应的业务模块,以利用所述业务模块对所述目标请求进行处理,得到处理后的目标请求;最后从预先构建的第二连接池中确定出处于空闲状态的第二长连接链路,并通过所述第二长连接链路将处理后的目标请求发送至所述消息服务器。

本申请实施例通过第一连接池中处于空闲状态的第一长连接链路接收消息服务器发送的目标请求,并通过第二连接池中处于空闲状态的第二长连接链路返回处理后的目标请求,基于预先构建的连接池复用长连接链路,降低消息服务器压力,提高请求处理效率。

[0064] 在一些具体实施例中,所述消息处理模块12,具体包括:

[0065] 排序单元,用于如果所述目标请求不止一个,则获取各个所述目标请求对应的优先级,并按照各个所述目标请求的优先级对各个所述目标请求进行排序,以得到对应的请求序列;

[0066] 缓存单元,用于通过消息队列的方式对所述请求序列中的各个所述目标请求进行存储;

[0067] 确定单元,用于确定各个所述目标请求对应的标识码;

[0068] 分发单元,用于根据各个所述目标请求对应的所述标识码将所述请求序列中的各个所述目标请求依次分发至所述后端服务器中相应的业务模块;其中,所述标识码与所述业务模块之间存在唯一对应关系。

[0069] 进一步的,本申请实施例还提供了一种电子设备。图7是根据一示例性实施例示出的电子设备20结构图,图中的内容不能认为是对本申请的使用范围的任何限制。

[0070] 图7为本申请实施例提供的一种电子设备20的结构示意图。该电子设备20,具体可以包括:至少一个处理器21、至少一个存储器22、电源23、通信接口24、输入输出接口25和通信总线26。其中,所述存储器22用于存储计算机程序,所述计算机程序由所述处理器21加载并执行,以实现前述任一实施例公开的请求处理方法中的相关步骤。

[0071] 本实施例中,电源23用于为电子设备20上的各硬件设备提供工作电压;通信接口24能够为电子设备20创建与外界设备之间的数据传输通道,其所遵循的通信协议是能够适用于本申请技术方案的任意通信协议,在此不对其进行具体限定;输入输出接口25,用于获取外界输入数据或向外界输出数据,其具体的接口类型可以根据具体应用需要进行选取,在此不进行具体限定。

[0072] 另外,存储器22作为资源存储的载体,可以是只读存储器、随机存储器、磁盘或者光盘等,其上所存储的资源可以包括操作系统221、计算机程序222及数据223等,存储方式可以是短暂存储或者永久存储。

[0073] 其中,操作系统221用于管理与控制电子设备20上的各硬件设备以及计算机程序222,以实现处理器21对存储器22中海量数据223的运算与处理,其可以是Windows Server、Netware、Unix、Linux等。计算机程序222除了包括能够用于完成前述任一实施例公开的由电子设备20执行的请求处理方法的计算机程序之外,还可以进一步包括能够用于完成其他特定工作的计算机程序。数据223可以包括电子设备20收集到的目标请求。

[0074] 进一步的,本申请实施例还公开了一种存储介质,所述存储介质中存储有计算机程序,所述计算机程序被处理器加载并执行时,实现前述任一实施例公开的请求处理方法步骤。

[0075] 本说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其它实施例的不同之处,各个实施例之间相同或相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0076] 最后,还需要说明的是,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。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个…”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0077] 以上对本发明所提供的请求处理方法、装置、设备及存储介质进行了详细介绍,本文中应用了具体个例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

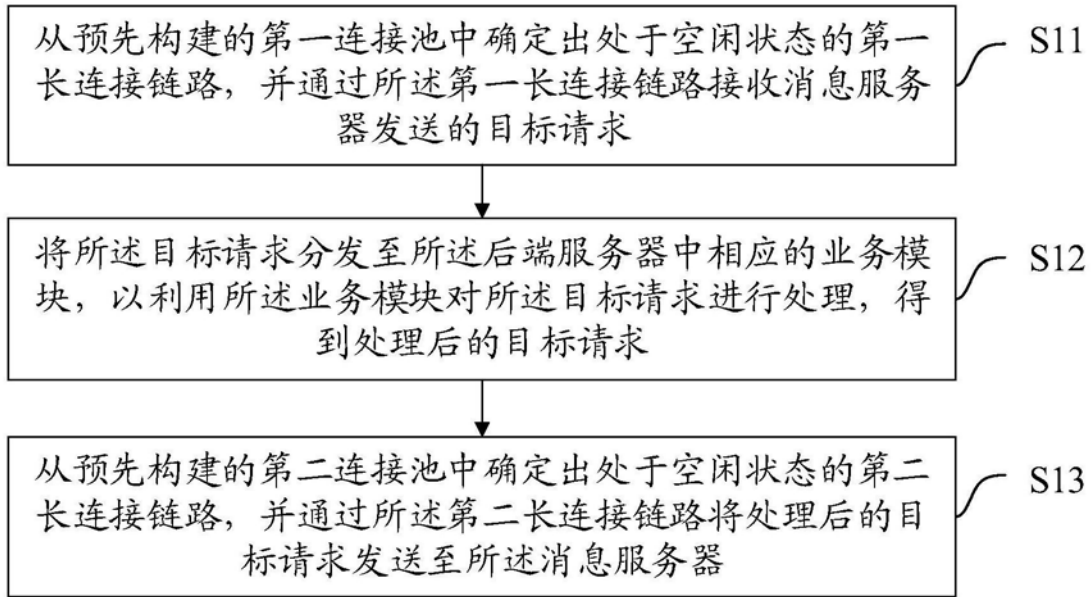


图1

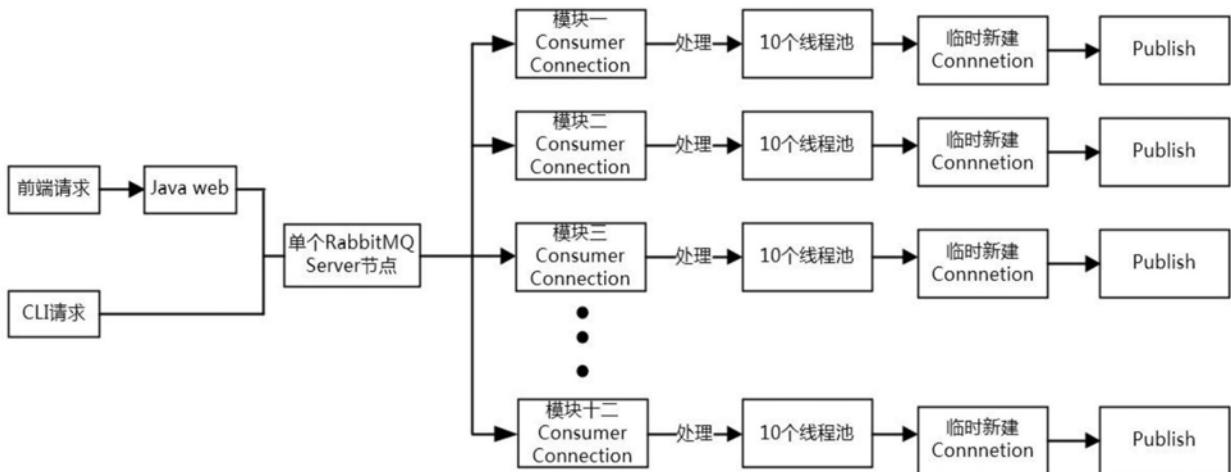


图2

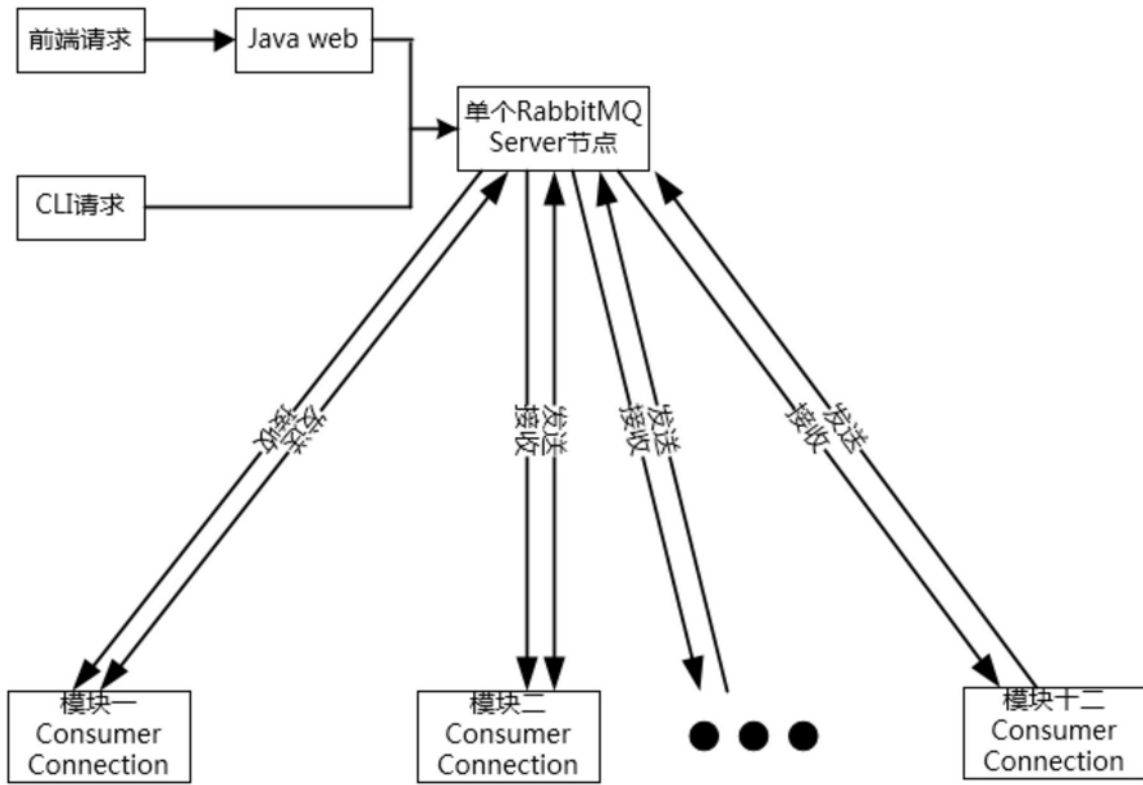


图3

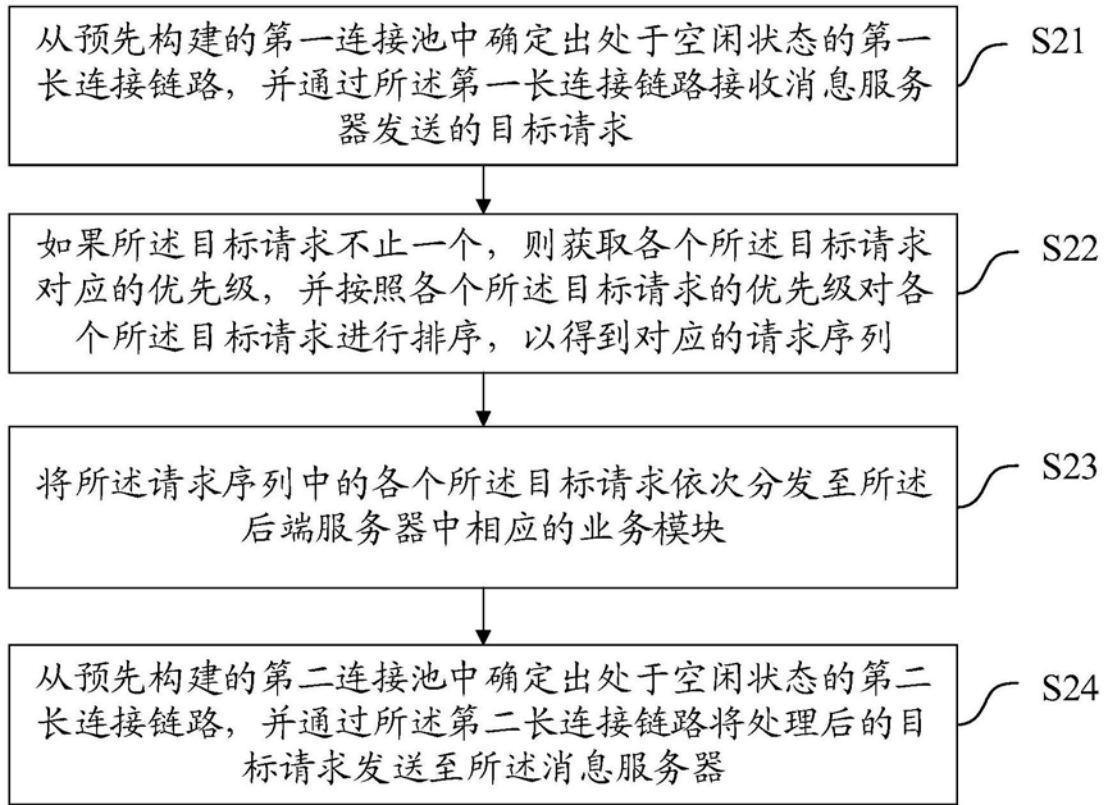


图4

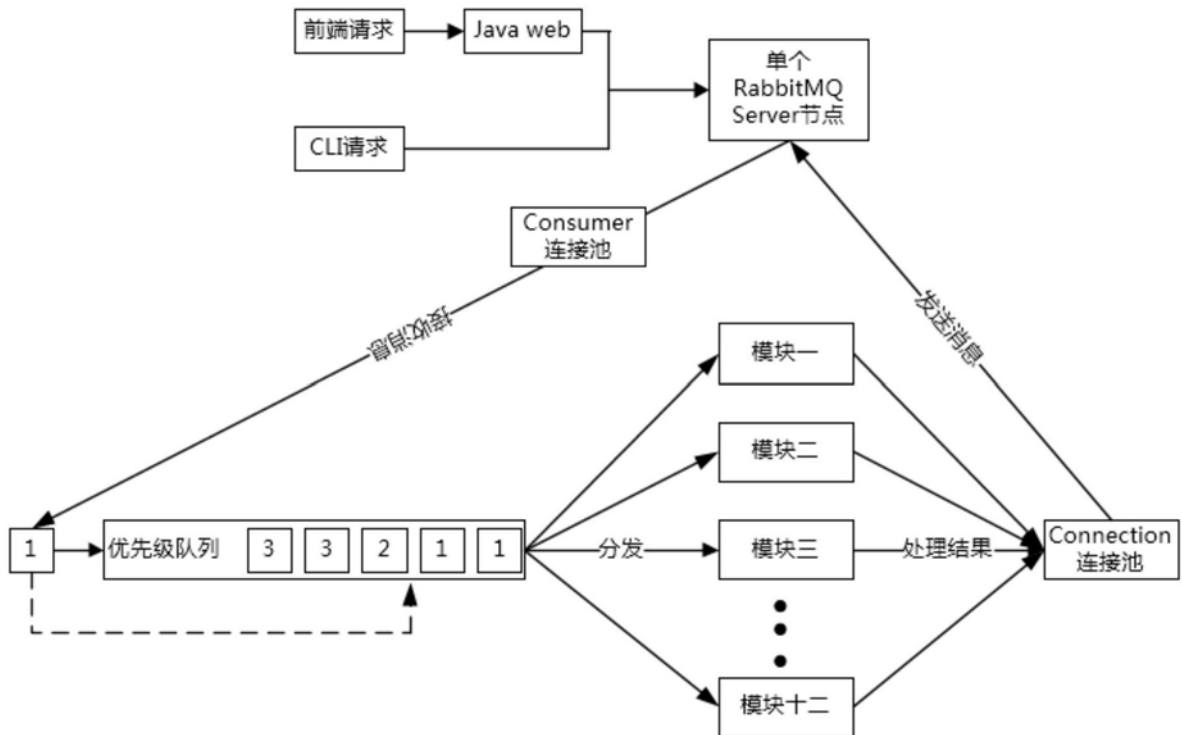


图5

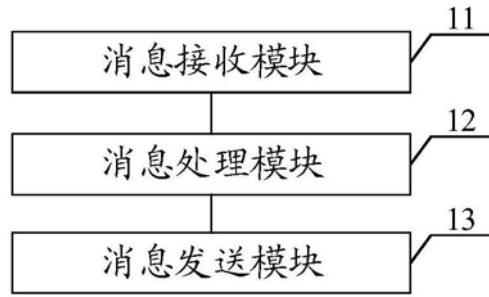


图6

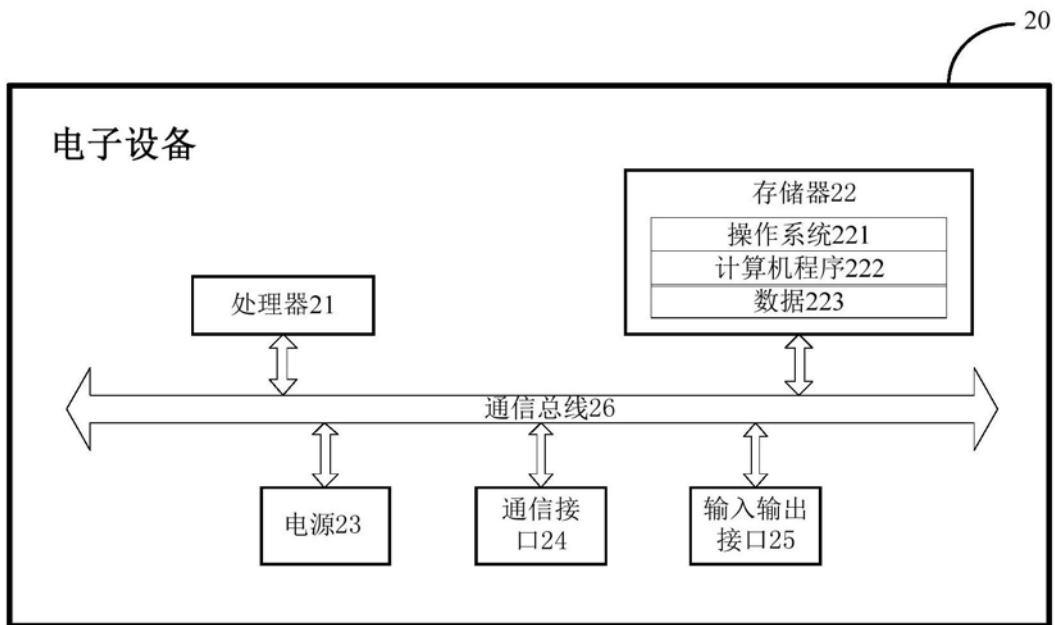


图7