资源调度系统及方法

本申请要求于 2022 年 01 月 29 日提交中国专利局、申请号为 202210111074.7、申请名称为“资源调度系统及方法”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本说明书实施例涉及计算机技术领域，特别涉及资源调度系统及方法。

背景技术

随着计算机技术的发展，资源调度器需要进行处理的资源调度请求的数量也与日俱增，由接口程序接收并存储业务管理器提交的资源调度请求，并依次将资源调度请求发送到资源调度器，由资源调度器针对资源调度请求进行处理，确定资源调度结果后，将资源调度结果发送到对应的业务管理器。

在同一时刻，接口程序接收到的资源调度请求数量较为庞大的情况下，则无法迅速的将资源调度请求发送到资源调度器，基于此，亟需一种资源调度方法解决这一问题。

发明内容

有鉴于此，本说明书实施例提供了一种资源调度系统。本说明书一个或者多个实施例同时涉及一种信息处理方法，一种资源调度方法，一种计算设备，一种计算机可读存储介质以及一种计算机程序，以解决现有技术中存在的技术缺陷。

根据本说明书实施例的第一方面，提供了一种资源调度系统，包括：

调度器和任务管理器，其中，所述任务管理器包括任务处理器和资源处理节点；

所述任务处理器，被配置为接收针对待处理任务提交的任务处理请求；基于所述任务处理请求创建资源获取请求并发送至所述调度器；

所述资源处理节点，被配置为接收所述任务处理请求；基于所述任务处理请求创建可调度资源信息并发送至所述调度器；

所述调度器，被配置为根据所述资源获取请求和所述可调度资源信息对所述待处理任务进行资源调度，获得资源调度结果；将所述资源调度结果发送至所述资源处理节点和所述任务处理器。

根据本说明书实施例的第二方面，提供了一种信息处理方法，应用于上述资源调度系统，包括：

任务管理器响应于业务请求确定业务信息，以及读取初始发送参数；基于所述业务信息将初始发送参数更新为目标发送参数；将所述业务信息和所述目标发送参数发送到调度器，并存储到发送信息队列；

所述调度器在接收到所述业务信息和所述目标发送参数的情况下，根据所述业务信息和所述目标发送参数确定反馈信息，以及读取初始接收参数；基于所述反馈信息将所述初始接收参数更新为目标接收参数，将所述反馈信息和所述目标接收参数发送到所述任务管理器；

所述任务管理器接收所述反馈信息和所述目标接收参数；基于所述目标接收参数更新所述业务信息对应的状态参数和所述目标发送参数，根据更新后的状态参数对所述发送信息队列进行更新。

根据本说明书实施例的第三方面，提供了一种资源调度方法，包括：

任务处理器接收针对待处理任务提交的任务处理请求；基于所述任务处理请求创建资源获取请求并发送至所述调度器；

资源处理节点接收所述任务处理请求；基于所述任务处理请求创建可调度资源信息并发送至所述调度器；

调度器根据所述资源获取请求和所述可调度资源信息对所述待处理任务进行资源调度，获得资源调度结果；将所述资源调度结果发送至所述资源处理节点和所述任务处理器。

根据本说明书实施例的第四方面，提供了一种计算设备，包括：

存储器和处理器；

所述存储器用于存储计算机可执行指令，所述处理器用于执行所述计算机可执行指令，该计算机可执行指令被处理器执行时实现上述资源调度方法或信息处理方法的步骤。

根据本说明书实施例的第五方面，提供了一种计算机可读存储介质，其存储有计算机可执行指令，该指令被处理器执行时实现上述资源调度方法或信息处理方法的步骤。

根据本说明书实施例的第六方面，提供了一种计算机程序，其中，当所述计算机程序在计算机中执行时，令计算机执行上述资源调度方法或信息处理方法的步骤。

本说明书提供的资源调度系统，在进资源调度时，由任务处理器接收针对待处理任务提交的任务处

理请求，基于任务处理请求创建资源获取请求并直接发送至调度器；资源处理节点在接收到任务处理请求后，基于任务处理请求创建可调度资源信息也发送至调度器；调度器根据资源获取请求和可调度资源信息对待处理任务进行资源调度，获得资源调度结果后将资源调度结果发送至资源处理节点和任务处理器，完成资源调度，由任务处理器和资源处理节点分别将资源获取请求和可调度资源信息直接发送到调度器，从而缩短了调度器获取资源获取请求和可调度资源信息的等待时间，提高任务处理请求的反馈速度。

附图说明

图 1 是本说明书一个实施例提供的一种资源调度系统的结构示意图；

图 2 是本说明书一个实施例提供的一种资源调度系统的处理过程示意图；

图 3 是本说明书一个实施例提供的一种资源调度系统的交互示意图；

图 4 是本说明书一个实施例提供的一种信息处理方法的流程图；

图 5 是本说明书一个实施例提供的一种信息处理方法的处理过程示意图；

图 6 是本说明书一个实施例提供的一种信息处理方法的交互示意图；

图 7 是本说明书一个实施例提供的一种资源调度方法的流程图；

图 8 是本说明书一个实施例提供的一种计算设备的结构框图。

具体实施方式

在下面的描述中阐述了很多具体细节以便于充分理解本说明书。但是本说明书能够以很多不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本说明书内涵的情况下做类似推广，因此本说明书不受下面公开的具体实施的限制。

在本说明书一个或多个实施例中使用的术语是仅仅出于描述特定实施例的目的，而非旨在限制本说明书一个或多个实施例。在本说明书一个或多个实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式，除非上下文清楚地表示其他含义。还应当理解，本说明书一个或多个实施例中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

应当理解，尽管在本说明书一个或多个实施例中可能采用术语第一、第二等来描述各种信息，但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如，在不脱离本说明书一个或多个实施例范围的情况下，第一也可以被称为第二，类似地，第二也可以被称为第一。取决于语境，如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

首先，对本说明书一个或多个实施例涉及的名词术语进行解释。

kubernetes，简称 K8s，是用 8 代替名字中间的 8 个字符“ubernete”而成的缩写。K8s 是一个开源的，用于管理云平台中多个主机上的容器化的应用，它支持自动化部署、大规模可伸缩、应用容器化管理。在 kubernetes 中，我们可以创建多个容器，每个容器里面运行一个应用实例，然后通过内置的负载均衡策略，实现对这一组应用实例的管理、发现、访问。

apiserver，用于暴露 kubernetes API。任何的资源请求/调用操作都是通过 apiserver 提供的接口进行。请参阅构建高可用集群。

appmaster，负责整个任务的运行，包括资源的申请、调度结果的接收，以及任务的发送等。

scheduler，监视新创建没有分配到机器节点的任务，为任务选择一个机器节点。

在本说明书中，提供了一种资源调度系统，本说明书同时涉及一种信息处理系统，一种资源调度方法，一种信息处理方法，一种计算设备，以及一种计算机可读存储介质，在下面的实施例中逐一进行详细说明。

在 kubernetes 框架中，现有技术通常是由任务管理器（appmaster）通过接口程序（apiserver）向调度器（scheduler）提交任务处理请求，调度器通过接口程序监听到任务处理请求后，为其产生资源调度结果，调度器将资源调度结果通过接口程序发送给提交任务处理请求的任务处理器，机器节点通过接口程序接收到任务处理请求对应的资源调度事件，任务处理器通过接口程序接收到任务处理请求对应的资源调度结果，由机器节点根据资源调度结果进行任务处理。然而接口程序为了保证接收到的任务处理请求的顺序，会对每个任务处理请求进行存储，串行的将任务处理请求发送到调度器，这种方法极易造成请求堆积，影响资源调度效率。

本说明说一个实施例提供的资源调度系统则是以 Kubernetes 框架为基础，对 Kubernetes 框架进行改进，由调度器直接与资源处理节点和任务处理器进行信息交互，从而降低了请求堆积的可能性，提高了资源调度的效率。本实施例以调度器在进行资源调度时为例，对资源调度系统进行资源调度处理的过程进行描述，相应的，其他资源调度方案均可参见本实施例相应的描述内容，在此不作过多赘述。

图 1 示出了本说明书一个实施例提供的一种资源调度系统的结构示意图，该资源调度系统 100 包括调度器 110 和任务管理器 120，其中，所述任务管理器 120 包括任务处理器 1202 和资源处理节点 1204，

其中,

所述任务处理器 1202，被配置为接收针对待处理任务提交的任务处理请求；基于所述任务处理请求创建资源获取请求并发送至所述调度器 110；

所述资源处理节点 1204，被配置为接收所述任务处理请求；基于所述任务处理请求创建可调度资源信息并发送至所述调度器 110；

所述调度器 110，被配置为根据所述资源获取请求和所述可调度资源信息对所述待处理任务进行资源调度，获得资源调度结果；将所述资源调度结果发送至所述资源处理节点 1204 和所述任务处理器 1202。

具体的，待处理任务是指客户端提交到资源调度系统的等待处理的作业任务；任务处理请求是指客户端所在的客户端向任务处理器 1202 发出的计算机指令，用于请求资源处理当前的作业任务；任务处理器 1202 用于接收待处理任务，根据待处理任务向调度器 110 请求分配用于执行待处理任务所需的资源，在 Kubernetes 框架中任务处理器即为 appmaster；资源获取请求是指任务处理器 1202 向调度器 110 发送的用于请求资源的计算机指令；资源处理节点 1204 用于接收待处理任务，并将当前可用资源的相关信息发送到调度器 110，在 Kubernetes 框架中调度器即为 scheduler；可调度资源信息即为当前可供调度器 110 进行资源调度的可用资源的相关信息；所述调度器 110 用于实现资源调度系统中资源的协调、分配和回收，根据特定的调度算法和策略对资源进行合理有效的分析和匹配，从而更合理与充分的利用资源，调度器 110 做为单独的程序运行，启动之后会不断的接收来自任务处理器 1202 的资源获取请求，和来自资源处理节点 1204 的可调度资源信息，对接收到的资源获取请求和可调度资源信息做出反馈，并将反馈结果发送到任务处理器 1202 和资源处理节点 1204，完成资源调度；任务管理器包括任务处理器和资源处理节点；资源调度即为调度器 110 基于接收到的资源获取请求和可调度资源信息确定反馈结果的过程；资源调度结果即为调度器 110 的反馈结果。

基于此，任务处理器 1202 接收到针对待处理任务提交的任务处理请求后，基于接收到的任务处理请求创建资源获取请求，并将资源获取请求发送到调度器 110；资源处理节点 1204 接收到任务处理请求后，基于任务处理请求创建可调度资源信息，并将可调度资源信息发送到调度器 110；调度器 110 接收到资源获取请求和可调度资源信息后对待处理任务进行资源调度，获得资源调度结果，并将资源调度结果分别发送至资源处理节点 1204 和任务处理器 1202。

综上所述，由任务处理器和资源处理节点分别将资源获取请求和可调度资源信息直接发送到调度器，从而缩短了调度器获取资源获取请求和可调度资源信息的等待时间，提高了任务处理请求的反馈速度。

进一步的，在任务处理器 1202 和资源处理节点 1204 接收到了调度器反馈的资源调度结果后，即可根据资源调度结果对待处理任务进行处理，本实施例中，所述任务处理器 1202，还被配置为接收所述资源调度结果，基于所述资源调度结果将所述待处理任务发送至所述资源处理节点 1204；所述资源处理节点 1204，还被配置为接收所述资源调度结果和所述待处理任务；根据所述资源调度结果确定所述待处理任务对应的目标资源，利用所述目标资源执行所述待处理任务。

具体的，目标资源是指任务处理器 1202 基于接收到的资源调度结果确定的，用于执行待处理任务的资源，当任务处理器 1202 根据资源调度结果，将待处理任务发送到资源处理节点 1204 后，资源处理节点 1204 即可利用与待处理任务匹配的资源进行任务处理。

基于此，在任务处理器 1202 接收到了调度器发送的资源调度结果后，基于接收到的资源调度结果将任务处理器 1202 中存储的待处理任务发送到资源处理节点 1204 进行执行。资源处理节点 1204 接收来自调度器的资源调度结果，同时接收来自任务处理器 1202 的待处理任务，根据接收到的调度器的资源调度结果在上报的可调度资源信息中确定目标资源对应的信息，利用目标资源对执行待处理任务。

进一步的，在资源处理节点 1204 完成了执行待处理任务后，为了提高资源的利用率，可以将用于执行待处理任务的目标资源释放，用于后续的任务处理，本实施例中，所述任务处理器 1202，还被配置为在监听到所述待处理任务执行完成的情况下，向所述调度器 110 发送与所述待处理任务对应的任务执行完成信息；

所述调度器 110，还被配置为基于所述任务执行完成信息创建资源释放请求，并将所述资源释放请求发送到所述资源处理节点 1204；所述资源处理节点 1204，还被配置为接收所述资源释放请求；根据所述资源释放请求释放所述目标资源。

具体的，任务执行完成信息是指当资源处理节点 1204 执行完成了待处理任务后，任务处理器 1202 监听到的信息，表示待处理任务已经完成；资源释放请求是指资源处理节点 1204 执行完成了待处理任务后，由调度器发送到资源处理节点 1204 的计算机指令，用于告知资源处理节点 1204 释放执行待处理任务时所占用的资源。

基于此，在任务处理器 1202 将待处理任务发送到资源处理节点 1204 进行处理后，不断的监听资源处理节点 1204 处的待处理任务的执行情况，在监听到了待处理任务执行完成的情况下，由任务处理

器 1202 向调度器发送与待处理任务对应的任务执行完成信息，调度器在接收到来自任务处理器 1202 的任务执行完成信息时，基于任务执行完成信息创建资源释放请求，用于告知资源处理节点 1204 释放其执行待处理任务时所占用的目标资源。由调度器将资源释放请求发送到资源处理节点 1204，资源处理节点 1204 接收到资源释放请求后，根据资源释放请求释放用于执行待处理任务的目标资源。

举例说明，如图 2 所示的示意图，调度器可以同时接收来自多个任务管理器的资源获取请求，每个任务管理器中同时存在多个待处理任务。调度器则基于接收到的多个资源获取请求分别进行处理，确定每个资源获取请求对应的资源调度结果，并将资源调度结果分别发送到任务管理器 1-3，由任务管理器将对应资源获取请求的资源调度结果分别发送到资源处理节点 1-3，再由资源处理节点发送到子资源处理节点进行资源确认和任务处理。在子资源处理节点的任务处理完成后，向调度器提交资源释放请求，释放任务处理时所占用的资源。

综上所述，调度器通过向资源处理节点发送资源释放请求实现对目标资源的释放，目标资源释放后，还可以用于执行其他任务，从而实现了目标资源的重复利用，提高了资源利用效率。

进一步的，在任务处理器 1202 和资源处理节点 1204 向调度器分别将资源获取请求和可调度资源信息时，由于网络、软件或硬件随时可能出现问题，则会导致资源调度系统出现故障，调度器则需要进行故障恢复，本实施例中，所述调度器 110，还被配置为在所述调度器 110 处于故障恢复状态的情况下，根据所述故障恢复状态创建信息获取请求；将所述信息获取请求发送至所述资源处理节点 1204 和所述任务处理器 1202；所述调度器 110，还被配置为在接收到所述信息获取请求关联的故障恢复信息的情况下，基于所述故障恢复信息进行故障恢复处理；所述调度器 110，还被配置为在未接收到所述故障恢复信息的情况下，将所述资源调度结果发送至所述资源处理节点 1204 和所述任务处理器 1202。

具体的，故障恢复状态是指计算机程序由于网络、进程重启、计算机硬件/软件等问题导致信息发送失败、通信中断后，进行网络恢复、问题修复的状态；信息获取请求是指在调度器发生故障后，需要向任务处理器 1202 和资源处理节点 1204 发送计算机指令，用于告知任务处理器 1202 和资源处理节点 1204 向调度器上报资源调度结果；故障恢复信息即为任务处理器 1202 和资源处理节点 1204 向调度器上报的资源调度结果；故障恢复处理是指调度器采用故障处理策略修复故障，恢复到正常状态的处理的过程。

基于此，在调度器将资源调度结果发送到资源处理节点 1204 和任务处理器 1202 时，在调度器产生故障，处于故障恢复状态的情况下，为了实现故障恢复，根据当前的故障恢复状态创建信息获取请求，并将信息获取请求发送到资源处理节点 1204 和任务处理器 1202，资源处理节点 1204 和任务处理器 1202 根据接收到的信息获取请求确定故障恢复信息，并将向调度器发送故障恢复信息，在调度器接收到故障恢复信息的情况下，根据故障恢复信息进行故障恢复处理；在调度器未接收到故障恢复信息的情况下，将资源调度结果分别发送到资源处理节点 1204 和任务处理器 1202。

综上所述，在调度器发送故障的情况下，立即向资源处理节点和任务处理器发送信息获取请求，以用于接收故障恢复信息，利用故障恢复信息进行故障恢复，从而进行未完成的资源调度工作，确保对每个任务处理请求都进行资源调度，避免调度器接收了任务处理请求但并未处理的情况。

进一步的，在调度器接收到故障恢复信息的情况下，为了准确的进行故障恢复，还需要确定接收到的故障恢复信息是来自任务处理器 1202 还是来自资源处理节点 1204，并选择对应的故障恢复处理，在本实施例中，所述调度器 110，还被配置为在接收到所述资源处理节点 1204 反馈的故障恢复信息的情况下，根据所述故障恢复信息确定目标资源调度结果，将所述目标资源调度结果发送至所述任务处理器 1202 和所述资源处理节点 1204；或者，在接收到所述任务处理器 1202 反馈的故障恢复信息的情况下，向所述任务处理器 1202 发送调度结果取消信息。

具体的，目标资源调度结果是指在调度器接收到资源处理节点 1204 反馈的故障恢复信息后，重新进行资源调度后确定的，针对故障恢复信息的调度结果。

基于此，在调度器接收到资源处理节点 1204 反馈的故障恢复信息的情况下，则说明在调度器发生故障前，资源处理节点 1204 已经根据资源调度结果确定了对应的资源，当调度器发生故障后，调度器再基于接收到的故障恢复信息确定目标资源调度结果，并将目标资源调度结果分别发送到任务处理器 1202 和资源处理节点 1204，用于任务处理器 1202 和资源处理节点 1204 对待处理任务进行处理；在调度器接收到的故障恢复信息来自于任务处理器 1202 时，则说明在调度器发生故障前资源处理节点 1204 未接收到资源调度结果，调度器立即向任务处理器 1202 发送调度结果取消信息，用于告知任务处理器 1202 取消调度器发生故障前接收到的资源调度结果。

综上所述，调度器通过判断接收到的故障恢复信息来自任务处理器或资源处理节点，对应的采用不同的故障恢复方法，实现灵活的进行故障恢复。

进一步的，在调度器等待任务处理器 1202 和资源处理节点 1204 上报故障恢复信息时，考虑到资

源的利用率问题，需要为调度器预设一个固定的等待时间区间，超过等待时间调度器则不继续等待任务处理器 1202 和资源处理节点 1204 上报故障恢复信息，本实施例中，所述调度器 110，还被配置为判断预设的时间区间内是否接收到所述故障恢复信息；若是，执行基于所述故障恢复信息进行故障恢复处理步骤；

若否，向所述资源处理节点 1204 和所述任务处理器 1202 发送调度结果取消信息。

具体的，时间区间是指为调度器预先设定的一个时间范围，调度器在这个时间范围内等待任务处理器 1202 和资源处理节点 1204 上报故障恢复信息，调度器的等待时间超过这个时间范围，则不再继续等待；调度结果取消信息是指调度器的等待时间超过预设的时间区间后，向任务处理器 1202 和资源处理节点 1204 发送的信息，用于告知任务处理器 1202 和资源处理节点 1204，当前任务处理器 1202 和资源处理节点 1204 接收到的调度结果不可用。

基于此，在调度器等待任务处理器 1202 和资源处理节点 1204 上报故障恢复信息时，预设一个固定范围的时间区间，在此时间区间内等待任务处理器 1202 和资源处理节点 1204 上报故障恢复信息，若在此时间区间内接收到了故障恢复信息，则进行故障恢复处理；若在此时间区间内未接收到故障恢复信息，则向资源处理节点 1204 和任务处理器 1202 发送调度结果取消信息，用于告知任务处理器 1202 和资源处理节点 1204 取消接收到的资源调度结果，以便于后续任务处理器 1202 和资源处理节点 1204 重现根据待处理任务向调度器发送资源获取请求。

举例说明，在 k8s 框架下进行资源调度时，在调度器由于网络等原因出现故障并进行重启时，调度器向 AppMaster 和 Node 发送请求，用于告知 AppMaster 和 Node 上报从调度器接收到的资源信息。当 AppMaster 和 Node 没有重启时，调度器接收到 AppMaster 和 Node 上报的资源信息后，调度器根据接收到的资源信息进行重启，若只接收到 AppMaster 上报的资源信息，未接收到 Node 上报的资源信息，则说明调度器在出现故障前，成功的将资源调度信息发送到了 AppMaster，但向 Node 发送资源调度信息失败了，在故障恢复后向 AppMaster 发送取消信息，取消 AppMaster 接收到的资源调度信息；若接收到 Node 上报的资源信息，未接收到 AppMaster 上报的资源信息，则说明调度器在出现故障前，成功的将资源调度信息发送到了 Node，但向 AppMaster 发送资源调度信息失败了，在故障恢复后向 AppMaster 发送资源调度信息。当 AppMaster 和 Node 也重启时，在预设的时间范围内等待接收 AppMaster 和 Node 上报的资源信息，若超过时间范围后仍未接收到 AppMaster 和 Node 上报的资源信息，当故障恢复后，即便 AppMaster 和 Node 上报了资源调度信息，调度器也告知 AppMaster 和 Node 取消上报的资源信息。

综上所述，通过预设调度器的等待时间区间，使得调度器在固定的时间区间内进行等待，从而避免由于调度器长时间等待接收故障恢复信息造成的资源浪费，任务处理器和资源处理节点也可以根据调度器的反馈提高待处理任务的处理效率。

图 3 示出了本申请一个实施例提供的一种资源调度系统的交互示意图，其中，包括任务管理器 340 和调度器 350，调度器 350 包括任务处理器 342 和资源处理节点 344，具体包括以下步骤：

步骤 S302，任务处理器向调度器提交资源获取请求。

步骤 S304，资源处理节点向调度器提交可调度资源信息。

步骤 S306，调度器根据资源获取请求和可调度资源信息进行资源调度，获得资源调度结果。

步骤 S308，调度器将资源调度结果发送到任务处理器。

步骤 S310，调度器将资源调度结果发送到资源处理节点。

步骤 S312，任务处理器基于资源调度结果将待处理任务发送到资源处理节点。

步骤 S314，资源处理节点接收资源调度结果，基于资源调度结果确定目标资源。

步骤 S316，资源处理节点接收待处理任务，基于目标资源执行待处理任务。

步骤 S318，任务处理器监听到待处理任务的执行完成后，向调度器发送任务执行完成信息。

步骤 S320，调度器根据任务执行完成信息向资源处理节点发送资源释放请求。

步骤 S322，资源处理节点释放目标资源。

综上所述，由任务处理器和资源处理节点分别将资源获取请求和可调度资源信息直接发送到调度器，从而缩短了调度器获取资源获取请求和可调度资源信息的等待时间，提高了任务处理请求的反馈速度。

图 4 示出了本说明书一个实施例提供的一种信息处理方法的流程图，该信息处理方法应用于上述资源调度系统，具体包括以下步骤：

步骤 S402，任务管理器响应于业务请求确定业务信息，以及读取初始发送参数；基于所述业务信息将初始发送参数更新为目标发送参数；将所述业务信息和所述目标发送参数发送到调度器，并存储到发送信息队列。

具体的，业务请求是指任务管理器接收到的计算机指令，用于向调度器请求资源处理当前的作业任务；业务信息是指与业务请求对应的资源需求信息，或用于发送到信息接收方，并等待信息接收方反馈

的信息；初始发送参数是指在任务管理器未确定业务信息之前系统当前的参数值；目标发送参数是指任务管理器确定了将要发送到调度器的一条业务信息后，初始发送参数更新后得到的参数值；发送信息队列是指用于存储业务信息和目标发送参数的线性表。

基于此，任务管理器在接收到业务请求后，响应于接收到的业务请求确定与业务请求对应的业务信息，并读取当前状态下的初始发送参数，基于确定的将要发送到调度器的业务信息对当前状态下的初始发送参数进行更新，将初始发送参数更新为目标发送参数，再将业务信息和目标发送参数一同发送到调度器，同时将业务信息和目标发送参数存储到任务管理器的发送信息队列中，以便于后续当业务信息和目标发送参数发送失败时，再次将业务信息和目标发送参数发送到调度器。

举例说明，如图 5 所示，当客户端有存储数据和删除数据的需求时，向资源管理器发送资源存储请求和资源删除请求，同时将资源存储请求和资源删除请求以及对应的发送参数存储到发送队列中，以便于后续在接收到调度器反馈的反馈信息和目标接收参数后对状态参数和目标发送参数进行更新。

步骤 S404，所述调度器在接收到所述业务信息和所述目标发送参数的情况下，根据所述业务信息和所述目标发送参数确定反馈信息，以及读取初始接收参数；基于所述反馈信息将所述初始接收参数更新为目标接收参数，将所述反馈信息和所述目标接收参数发送到所述任务管理器。

具体的，在上述任务管理器向调度器发送了业务信息和目标发送参数后，调度器则需要在接收到业务信息和目标发送参数后向任务管理器发送反馈信息，并更新初始接收参数，其中反馈信息是指调度器接收到任务管理器发送的业务信息和目标发送参数后确定的信息；初始接收参数在调度器未接收到业务信息之前，作为信息的接收方，系统当前的参数值；目标接收参数是指调度器确定了反馈信息后初始接收参数更新后得到的参数值。

基于此，调度器在接收到任务管理器发送的业务信息和目标发送参数的情况下，根据接收到的业务信息和目标发送参数确定与业务信息对应的反馈信息，读取调度器当前状态下对应的初始接收参数，根据确定的与业务信息对应的反馈信息对初始接收参数进行更新，将初始接收参数更新为目标接收参数，并将反馈信息和由初始接收参数更新得到的目标接收参数一同发送到任务管理器，同时将反馈信息和目标接收参数一同存储到调度器对应的发送队列中，以便于后续当未接收到任务管理器发送的业务信息时，再次向任务管理器发送反馈信息和目标接收参数。

进一步的，在调度器接收到业务信息和目标发送参数后，在将反馈信息发送到任务管理器之前，为了提高信息发送的可靠性，还需要基于反馈信息对初始接收参数进行更新，具体实现如下：

所述调度器读取初始接收参数，并将所述初始接收参数更新为中间接收参数；基于所述反馈信息将所述中间接收参数更新为目标接收参数。

具体的，中间接收参数是指当调度器接收到了来自任务管理器的业务信息后，对初始接收参数进行更新后的结果，用于表示调度器确认接收到了一条来自任务管理器的业务信息。

基于此，在调度器接收到业务信息和目标发送参数后，调度器读取系统当前的初始接收参数，由于接收到了一条业务信息，则基于随业务信息一同接收到的目标发送参数对初始接收参数进行更新，将初始接收参数更新为中间接收参数，当调度器确认了将要发送到任务管理器的反馈信息后，基于反馈信息对中间接收参数进行更新，将中间接收参数更新为目标接收参数。

综上所述，通过根据目标发送参数确定中间接收参数的方法，实现将初始接收参数更新为中间接收参数，提高了初始接收参数进行更新的准确性，同时当调度器确认了业务信息，任务管理器确认了反馈信息时，分别对初始发送参数和中间接收参数进行更新，从而提高了信息发送的可靠性。

进一步的，在任务管理器确定了业务信息后，为了提高信息发送的可靠性以及记录信息的发送顺序，还需要基于业务信息对初始发送参数进行更新，以及在调度器接收到目标发送参数后，基于目标发送参数对初始接收参数进行更新，具体实现如下：

所述任务管理器基于所述业务信息对初始发送参数中包含的发送信息参数进行更新，获得目标发送参数；所述调度器，还被配置为根据所述目标发送参数中包含的发送信息参数，对所述初始接收参数中包含的第一确认参数进行更新，获得中间接收参数。

具体的，发送信息参数是指初始发送参数中包含的参数，用于表示任务管理器向调度器发送了业务信息；第一确认参数是指初始接收参数中包含的参数，用于表示调度器接收到了任务管理器发送的业务信息，以及包含更新后的发送信息参数的初始发送参数。

基于此，在任务管理器确定了业务信息后，基于确定的业务信息对初始发送参数中包含的发送信息参数进行更新，获得目标发送参数，将业务信息和目标发送参数一同发送到调度器。调度器接收到任务管理器发送的业务信息和目标发送参数后，根据接收到的目标发送参数中包含的发送信息参数，对初始接收参数中包含的第一确认参数进行更新，获得中间接收参数。

综上所述，通过根据确定的业务信息更新初始发送参数，根据发送信息参数更新初始接收参数，提

高了初始发送参数和初始接收参数更新的准确性。

进一步的，在调度器根据目标发送参数将初始接收参数更新为中间接收参数后，在调度器发送反馈信息前，为了提高信息发送的可靠性以及记录信息的发送顺序，还需要对中间接收参数进行更新，具体实现如下：

所述调度器根据所述反馈信息和所述中间接收参数对所述中间接收参数包含的接收信息参数进行更新，获得目标接收参数。

具体的，接收信息参数是指中间接收参数中包含的参数，用于表示调度器接收到了任务管理器发送的业务信息和目标发送参数，并确定了反馈信息，确定了将要发送到业务管理器的反馈信息后，即对接收信息参数进行更新。

基于此，在调度器接收到业务管理器发送的业务信息和目标发送参数后，基于接收到的业务信息确定将要反馈给业务管理器的反馈信息，调度器根据已经确定的反馈信息和中间接收参数对调度器的中间接收参数包含的接收信息参数进行更新，获得目标接收参数。

综上所述，通过根据确定的反馈信息将中间接收参数更新为目标接收参数，提高了中间接收参数更新的准确性。

进一步的，在任务管理器接收到调度器的反馈信息和目标接收参数后，则说明任务管理器接收到了调度器针对业务信息的反馈结果，即调度器对业务信息的回复，此时还需要将发生信息队列中存储的业务信息和目标发送参数一同删除，并对目标发送参数进行更新，具体实现如下：

所述任务管理器在接收到所述目标接收参数的情况下，基于所述目标接收参数中包含的所述接收信息参数更新所述目标发送参数中包含的第二确认参数和所述状态参数；基于更新后的状态参数对所述发送信息队列中存储的所述业务信息和所述目标发送参数进行删除。

具体的，第二确认参数是指目标发送参数中包含的参数，用于表示任务管理器接收到了调度器发送的，针对业务信息的反馈信息；所述状态参数具体是指一个逻辑值，具有 TRUE 和 FALSE，0 和 1 两种情况，用于表示当任务管理器或调度器发出信息后，是否接收到反馈信息，若是则状态参数为 TRUE，若否则状态参数为 FALSE。

基于此，在任务管理器接收到调度器发送的目标接收参数的情况下，基于接收到的目标接收参数中包含的接收信息参数对目标发送参数中包含的第二确认参数进行更新，同时更新任务管理器的状态参数；在状态参数更新后，对任务管理器的发送信息队列中存储的业务信息和目标发送参数进行删除，以便于释放业务信息和目标发送参数所占用的存储空间。

举例说明，如图 5 所示，当客户端向资源管理器发送资源存储请求和资源删除请求，同时将资源存储请求和资源删除请求以及对应的发送参数存储到发送队列中，资源管理器接收到资源存储请求和资源删除请求时，资源管理器对应的接收参数进行更新，同时确定存储失败和删除失败的反馈信息，将反馈信息发送给客户端，同时更新资源管理器对应的接收参数，当客户端接收到反馈信息后，再次对发送参数进行更新，同时更新状态参数，表示接收到了资源管理器的回复。

此外，资源调度系统中任意两个节点之间的交互过程均可通过本实施例提供的信息处理方法实现，其他节点之间的交互过程均可参见本实施例相同或相应的描述内容，在此不作过多赘述。

综上所述，在任务管理器接收到反馈信息和目标接收参数后，对第二确认参数和状态参数进行更新，并删除发送信息队列中存储的业务信息和目标发送参数，提高了信息发送的可靠性，同时也提高了存储空间的利用率。

步骤 S406，所述任务管理器接收所述反馈信息和所述目标接收参数；基于所述目标接收参数更新所述业务信息对应的状态参数和所述目标发送参数，根据更新后的状态参数对所述发送信息队列进行更新。

具体的，在上述，调度器反馈了反馈信息并发送了目标接收参数后，任务管理器在接收到反馈信息和目标接收参数后，即可更新与业务信息对应的状态参数和目标发送参数，其中，状态参数用来表明任务管理器发送业务信息后是否收到调度器的反馈信息，状态参数用布尔值表示，即，若是，状态参数为 TRUE，若否，状态参数为 FALSE。

基于此，任务管理器接收到来自调度器的反馈信息和目标接收参数后，根据接收到的目标接收参数对业务信息对应的状态参数和目标发送参数进行更新，根据更新后的状态参数对任务管理器的发送信息队列进行更新，即，将发送队列中存储的业务信息进行删除，从而节省发送队列的空间。

进一步的，考虑到在任务管理器和调度器之间进行信息交互时，由于网络等多种原因，可能出现任务管理器向调度器发送了信息，但是调度器并未收到的情况，或调度器发送了信息但任务管理器并未收到的情况，因此还需要任务管理器再次将业务信息和目标发送参数发送到调度器，具体实现如下：

所述任务管理器在未接收到所述反馈信息的情况下，按照预设的信息发送规则将所述发送信息队列

中存储的所述业务信息和所述目标发送参数发送到所述调度器。

具体的，信息发送规则是指针对发送信息队列预设的规矩和准则，在本实施例中，信息发送规则可以是在预设的时间范围内，间隔固定的时间周期将发送信息队列中存储的业务信息和目标发送参数发送到调度器。

基于此，在任务管理器将业务信息和目标发送参数发送到调度器后，在任务管理器未接收到调度器发送的反馈信息的情况下，在预设的时间范围内，按照固定的时间间隔将发送信息队列中存储的业务信息和目标发送参数发送到调度器，并等待接收调度器的反馈信息。

相应的，在调度器向任务管理器发送了反馈信息后，未接收到任务管理器再次发送的业务信息的情况下，在预设的时间范围内，按照固定的时间间隔将接收信息队列中存储的反馈信息和目标接收参数发送到任务管理器，并等待接收任务管理器的反馈信息。

沿用上例，在客户端向资源管理器发送资源存储请求和资源删除请求后，未接收到反馈信息的情况下，在预设的时间范围内，按照固定的时间间隔将发送队列中存储的资源存储请求和资源删除请求发送到资源管理器，等待资源管理器的反馈信息。

综上所述，在任务管理器未接收到反馈信息的情况下，通过重新向调度器发送业务信息和目标发送参数，即便由于网络等因素导致了调度器未接收到业务信息和目标发送参数，也可以在任务管理器重发的情况下接收到业务信息和目标发送参数，从而提高信息发送的可靠性。

图 6 示出了本申请一个实施例提供的一种信息处理方法的交互示意图，具体包括以下步骤：

步骤 S602，任务管理器基于业务请求确定业务信息 A，读取发送参数 S={sendId_s=0，ackId_s=0}。

步骤 S604，任务管理器基于业务信息 A 对发送参数 S 进行更新，得到发送参数 S1={sendId_s=1，ackId_s=0}，将业务信息 A 和发送参数 S1 发送到调度器，并存储到发送方信息队列；其中，业务信息 A 的状态值为 F。

步骤 S606，任务管理器在网络故障的情况下，在预设的时间内重复将业务信息 A 和发送参数 S1 发送到调度器。

步骤 S608，任务管理器在预设的时间外基于业务信息 B 对发送参数 S1 进行更新，得到发送参数 S2={sendId_s=2，ackId_s=0}，将业务信息 B 和发送参数 S2 发送到调度器，并存储到发送方信息队列；其中，业务信息 B 的状态值为 F。

步骤 S610，任务管理器在预设的时间内未接收到反馈信息的情况下，重复将业务信息 A、业务信息 B 和发送参数 S2 发送到调度器。

步骤 S612，调度器在接收到业务信息 A、业务信息 B 和发送参数 S2 的情况下，基于业务信息 A、业务信息 B 和发送参数 S2，确定业务信息 A 对应的反馈信息 A，以及业务信息 B 对应的反馈信息 B，读取接收参数 R={sendId_r=0，ackId_r=0}，基于发送参数 S2 对接收参数 R 进行更新，得到接收参数 R1={sendId_r=0，ackId_r=2}，其中，反馈信息 A 和反馈信息 B 的状态值均为 F。

步骤 S614，调度器将接收参数 R1 更新为接收参数 R2={sendId_r=2，ackId_r=2}，并将反馈信息 A、反馈信息 B 和接收参数 R2 发送到任务管理器，并存储到接收方信息队列，其中，反馈信息 A 和反馈信息 B 的状态值均为 F。

步骤 S616，任务管理器在因为网络故障问题未接收到反馈信息 A 和反馈信息 B 的情况下，基于业务信息 C 对发送参数 S2 进行更新，得到发送参数 S3={sendId_s=3，ackId_s=0}，将业务信息 C 和发送参数 S3 发送到调度器，并存储到发送方信息队列；其中，业务信息 C 的状态值为 F。

步骤 S618，调度器在网络故障解决后接收业务信息 C 和发送参数 S3，确定业务信息 C 对应的反馈信息 C，基于发送参数 S3 对接收参数 R2 进行更新，得到接收参数 R3={sendId_r=2，ackId_r=3}其中，反馈信息 C 的状态值为 F。

步骤 S620，任务管理器在网络故障解决后接收到反馈信息 A、反馈信息 B、和接收参数 R2；基于接收参数 R2 中的 ackId_r=2 对发送参数 S3 进行更新，得到发送参数 S4={sendId_s=3，ackId_s=2}；将发送方信息队列中存储的业务信息 A 和业务信息 B 删除，同时将业务信息 A 和业务信息 B 状态值更新为 T。

其中，步骤 S618 与步骤 S620 在网络故障接触后，同时执行。

步骤 S622，任务管理器基于业务信息 D 对发送参数 S4 进行更新，得到发送参数 S5={sendId_s=4，ackId_s=2}，将业务信息 D 和发送参数 S5 发送到调度器，并存储到发送方信息队列；其中，业务信息 D 的状态值为 F。

步骤 S624，调度器在接收到业务信息 D 和发送参数 S5 的情况下，确定业务信息 D 对应的反馈信息 D，基于发送参数 S5 对接收参数 R3 进行更新，得到接收参数 R4={sendId_r=2，ackId_r=4}。

步骤 S626，调度器将接收参数 R4 更新为接收参数 R5={sendId_r=3，ackId_r=4}，并将反馈信息 C 和接收参数 R5 发送到任务管理器，并存储到接收方信息队列，其中，状态值为 F。

其中，步骤 S624 与步骤 S626 同时执行；以此类推，直至交互完成即可。

综上所述，在任务管理器和调度器之间进行信息交互的同时，维护任务管理器的发送参数和调度器接收参数，从而避免任务管理器和调度器之间互相发送多条信息时信息的接收顺序出现混乱，同时确保任务管理器或调度器发出的每一条信息都得到反馈。

与上述系统实施例相对应，本说明书还提供了一种资源调度方法，图 7 示出了本说明书一个实施例提供的一种资源调度方法的流程图，具体包括以下步骤：

步骤 702，所述任务处理器接收针对待处理任务提交的任务处理请求；基于所述任务处理请求创建资源获取请求并发送至所述调度器。

步骤 704，所述资源处理节点接收所述任务处理请求；基于所述任务处理请求创建可调度资源信息并发送至所述调度器。

步骤 706，所述调度器根据所述资源获取请求和所述可调度资源信息对所述待处理任务进行资源调度，获得资源调度结果；将所述资源调度结果发送至所述资源处理节点和所述任务处理器。

一个可选的实施例中，所述资源调度方法，包括：

调度器在所述调度器处于故障恢复状态的情况下，根据所述故障恢复状态创建信息获取请求；将所述信息获取请求发送至所述资源处理节点和所述任务处理器；调度器在接收到所述信息获取请求关联的故障恢复信息的情况下，基于所述故障恢复信息进行故障恢复处理；调度器在未接收到所述故障恢复信息的情况下，将所述资源调度结果发送至所述资源处理节点和所述任务处理器。

一个可选的实施例中，所述资源调度方法，包括：

调度器在接收到所述资源处理节点反馈的故障恢复信息的情况下，根据所述故障恢复信息确定目标资源调度结果，将所述目标资源调度结果发送至所述任务处理器和所述资源处理节点；或者，在接收到所述任务处理器反馈的故障恢复信息的情况下，向所述任务处理器发送调度结果取消信息。

一个可选的实施例中，所述资源调度方法，包括：

调度器判断预设的时间区间内是否接收到所述故障恢复信息；若是，执行基于所述故障恢复信息进行故障恢复处理步骤；若否，向所述资源处理节点和所述任务处理器发送调度结果取消信息。

一个可选的实施例中，所述资源调度方法，包括：

任务处理器接收所述资源调度结果，基于所述资源调度结果将所述待处理任务发送至所述资源处理节点；资源处理节点接收所述资源调度结果和所述待处理任务；根据所述资源调度结果确定所述待处理任务对应的目标资源，利用所述目标资源执行所述待处理任务。

一个可选的实施例中，所述资源调度方法，包括：

任务处理器在监听到所述待处理任务执行完成的情况下，向所述调度器发送与所述待处理任务对应的任务执行完成信息；调度器基于所述任务执行完成信息创建资源释放请求，并将所述资源释放请求发送到所述资源处理节点；资源处理节点接收所述资源释放请求；根据所述资源释放请求释放所述目标资源。

上述为本实施例的资源调度方法和信息处理方法的示意性方案。需要说明的是，该资源调度方法的技术方案与上述的资源调度系统的技术方案属于同一构思，信息处理方法的技术方案与上述的信息处理系统的技术方案属于同一构思，资源调度方法的技术方案未详细描述的细节内容，均可以参见上述资源调度系统的技术方案的描述，信息处理方法的技术方案未详细描述的细节内容，均可以参见上述信息处理系统的技术方案的描述。

图 8 示出了根据本说明书一个实施例提供的一种计算设备 800 的结构框图。该计算设备 800 的部件包括但不限于存储器 810 和处理器 820。处理器 820 与存储器 810 通过总线 830 相连接，数据库 850 用于保存数据。

计算设备 800 还包括接入设备 840，接入设备 840 使得计算设备 800 能够经由一个或多个网络 860 通信。这些网络的示例包括公用交换电话网（PSTN）、局域网（LAN）、广域网（WAN）、个域网（PAN）或诸如因特网的通信网络的组合。接入设备 840 可以包括有线或无线的任何类型的网络接口（例如，网络接口卡（NIC））中的一个或多个，诸如 IEEE802.11 无线局域网（WLAN）无线接口、全球微波互联接入（Wi-MAX）接口、以太网接口、通用串行总线（USB）接口、蜂窝网络接口、蓝牙接口、近场通信（NFC）接口，等等。

在本说明书的一个实施例中，计算设备 800 的上述部件以及图 8 中未示出的其他部件也可以彼此相连接，例如通过总线。应当理解，图 8 所示的计算设备结构框图仅仅是出于示例的目的，而不是对本说明书范围的限制。本领域技术人员可以根据需要，增添或替换其他部件。

计算设备 800 可以是任何类型的静止或移动计算设备，包括移动计算机或移动计算设备（例如，平板计算机、个人数字助理、膝上型计算机、笔记本计算机、上网本等）、移动电话（例如，智能手机）、

可佩戴的计算设备（例如，智能手表、智能眼镜等）或其他类型的移动设备，或者诸如台式计算机或PC的静止计算设备。计算设备800还可以是移动式或静止式的服务器。

其中，处理器820用于执行如下计算机可执行指令，该计算机可执行指令被处理器执行时实现上述方法的步骤。

上述为本实施例的一种计算设备的示意性方案。需要说明的是，该计算设备的技术方案与上述的方法的技术方案属于同一构思，计算设备的技术方案未详细描述的细节内容，均可以参见上述方法的技术方案的描述。

本说明书一个实施例还提供一种计算机可读存储介质，其存储有计算机可执行指令，该计算机可执行指令被处理器执行时实现上述方法的步骤。

上述为本实施例的一种计算机可读存储介质的示意性方案。需要说明的是，该存储介质的技术方案与上述的方法的技术方案属于同一构思，存储介质的技术方案未详细描述的细节内容，均可以参见上述方法的技术方案的描述。

本说明书一个实施例还提供一种计算机程序，其中，当所述计算机程序在计算机中执行时，令计算机执行上述方法的步骤。

上述为本实施例的一种计算机程序的示意性方案。需要说明的是，该计算机程序的技术方案与上述的方法的技术方案属于同一构思，计算机程序的技术方案未详细描述的细节内容，均可以参见上述方法的技术方案的描述。

上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

所述计算机指令包括计算机程序代码，所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括：能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器（ROM，Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、电载波信号、电信信号以及软件分发介质等。需要说明的是，所述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减，例如在某些司法管辖区，根据立法和专利实践，计算机可读介质不包括电载波信号和电信信号。

需要说明的是，对于前述的各方法实施例，为了简便描述，故将其都表述为一系列的动作组合，但是本领域技术人员应该知悉，本说明书实施例并不受所描述的动作顺序的限制，因为依据本说明书实施例，某些步骤可以采用其它顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于优选实施例，所涉及的动作和模块并不一定都是本说明书实施例所必须的。

在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其它实施例的相关描述。

以上公开的本说明书优选实施例只是用于帮助阐述本说明书。可选实施例并没有详尽叙述所有的细节，也不限制该发明仅为所述的具体实施方式。显然，根据本说明书实施例的内容，可作很多的修改和变化。本说明书选取并具体描述这些实施例，是为了更好地解释本说明书实施例的原理和实际应用，从而使所属技术领域技术人员能很好地理解和利用本说明书。本说明书仅受权利要求书及其全部范围和等效物的限制。

权 利 要 求 书

1.一种资源调度系统，包括：

调度器和任务管理器，其中，所述任务管理器包括任务处理器和资源处理节点；

所述任务处理器，被配置为接收针对待处理任务提交的任务处理请求；基于所述任务处理请求创建资源获取请求并发送至所述调度器；

所述资源处理节点，被配置为接收所述任务处理请求；基于所述任务处理请求创建可调度资源信息并发送至所述调度器；

所述调度器，被配置为根据所述资源获取请求和所述可调度资源信息对所述待处理任务进行资源调度，获得资源调度结果；将所述资源调度结果发送至所述资源处理节点和所述任务处理器。

2.根据权利要求 1 所述的资源调度系统，所述调度器，还被配置为在所述调度器处于故障恢复状态的情况下，根据所述故障恢复状态创建信息获取请求；将所述信息获取请求发送至所述资源处理节点和所述任务处理器；

所述调度器，还被配置为在接收到所述信息获取请求关联的故障恢复信息的情况下，基于所述故障恢复信息进行故障恢复处理；

所述调度器，还被配置为在未接收到所述故障恢复信息的情况下，将所述资源调度结果发送至所述资源处理节点和所述任务处理器。

3.根据权利要求 2 所述的资源调度系统，所述调度器，还被配置为在接收到所述资源处理节点反馈的故障恢复信息的情况下，根据所述故障恢复信息确定目标资源调度结果，将所述目标资源调度结果发送至所述任务处理器和所述资源处理节点；或者，

在接收到所述任务处理器反馈的故障恢复信息的情况下，向所述任务处理器发送调度结果取消信息。

4.根据权利要求 2 所述的资源调度系统，所述调度器，还被配置为判断预设的时间区间内是否接收到所述故障恢复信息；

若是，执行基于所述故障恢复信息进行故障恢复处理步骤；

若否，向所述资源处理节点和所述任务处理器发送调度结果取消信息。

5.根据权利要求 1 所述的资源调度系统，所述任务处理器，还被配置为接收所述资源调度结果，基于所述资源调度结果将所述待处理任务发送至所述资源处理节点；

所述资源处理节点，还被配置为接收所述资源调度结果和所述待处理任务；根据所述资源调度结果确定所述待处理任务对应的目标资源，利用所述目标资源执行所述待处理任务。

6.根据权利要求 5 所述的资源调度系统，所述任务处理器，还被配置为在监听到所述待处理任务执行完成的情况下，向所述调度器发送与所述待处理任务对应的任务执行完成信息；

所述调度器，还被配置为基于所述任务执行完成信息创建资源释放请求，并将所述资源释放请求发送到所述资源处理节点；

所述资源处理节点，还被配置为接收所述资源释放请求；根据所述资源释放请求释放所述目标资源。

7.一种信息处理方法，应用于权利要求 1-6 任意一项所述的资源调度系统，包括：

任务管理器响应于业务请求确定业务信息，以及读取初始发送参数；基于所述业务信息将初始发送参数更新为目标发送参数；将所述业务信息和所述目标发送参数发送到调度器，并存储到发送信息队列；

所述调度器在接收到所述业务信息和所述目标发送参数的情况下，根据所述业务信息和所述目标发送参数确定反馈信息，以及读取初始接收参数；基于所述反馈信息将所述初始接收参数更新为目标接收参数，将所述反馈信息和所述目标接收参数发送到所述任务管理器；

所述任务管理器接收所述反馈信息和所述目标接收参数；基于所述目标接收参数更新所述业务信息对应的状态参数和所述目标发送参数，根据更新后的状态参数对所述发送信息队列进行更新。

8.根据权利要求 7 所述的信息处理方法，所述调度器读取初始接收参数，并将所述初始接收参数更新为中间接收参数；基于所述反馈信息将所述中间接收参数更新为目标接收参数。

9.根据权利要求 8 所述的信息处理方法，所述任务管理器基于所述业务信息对初始发送参数中包含的发送信息参数进行更新，获得目标发送参数；

所述调度器，还被配置为根据所述目标发送参数中包含的发送信息参数，对所述初始接收参数中包含的第一确认参数进行更新，获得中间接收参数。

10.根据权利要求 9 所述的信息处理方法，所述调度器根据所述反馈信息和所述中间接收参数对所述中间接收参数包含的接收信息参数进行更新，获得目标接收参数。

11.根据权利要求 10 所述的信息处理方法，所述任务管理器在接收到所述目标接收参数的情况下，基于所述目标接收参数中包含的所述接收信息参数更新所述目标发送参数中包含的第二确认参数和所

述状态参数；基于更新后的状态参数对所述发送信息队列中存储的所述业务信息和所述目标发送参数进行删除。

12.根据权利要求 7 所述的信息处理方法，所述任务管理器在未接收到所述反馈信息的情况下，按照预设的信息发送规则将所述发送信息队列中存储的所述业务信息和所述目标发送参数发送到所述调度器。

13.一种资源调度方法，包括：

任务处理器接收针对待处理任务提交的任务处理请求；基于所述任务处理请求创建资源获取请求并发送至所述调度器；

资源处理节点接收所述任务处理请求；基于所述任务处理请求创建可调度资源信息并发送至所述调度器；

调度器根据所述资源获取请求和所述可调度资源信息对所述待处理任务进行资源调度，获得资源调度结果；将所述资源调度结果发送至所述资源处理节点和所述任务处理器。

14.一种计算设备，包括：

存储器和处理器；

所述存储器用于存储计算机可执行指令，所述处理器用于执行所述计算机可执行指令，该计算机可执行指令被处理器执行时实现权利要求 7-13 任意一项所述方法的步骤。

15.一种计算机可读存储介质，其存储有计算机可执行指令，该计算机可执行指令被处理器执行时实现权利要求 7-13 任意一项所述方法的步骤。