



(12) 发明专利

(10) 授权公告号 CN 102279730 B

(45) 授权公告日 2014. 02. 05

(21) 申请号 201010200891. 7

JP 特开 2010-39526 A, 2010. 02. 18, 说明书
第 7、10-17、30、46 段, 附图 1、9.

(22) 申请日 2010. 06. 10

审查员 梁岩

(73) 专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼岛资本大厦一座
四层 847 号邮箱

(72) 发明人 岑文初

(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 逯长明 王宝筠

(51) Int. Cl.

G06F 9/38 (2006. 01)

G06F 9/50 (2006. 01)

(56) 对比文件

CN 101561767 A, 2009. 10. 21, 说明书第 4 页
最后 1 段.

CN 101566957 A, 2009. 10. 28, 全文.

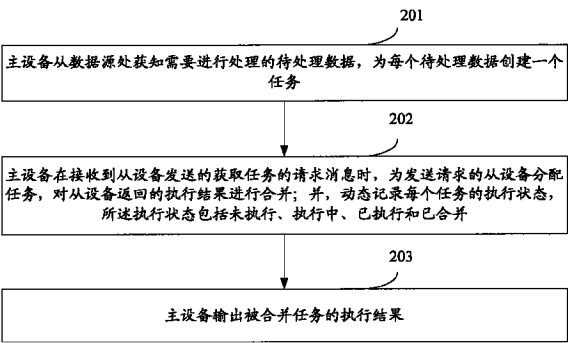
权利要求书2页 说明书7页 附图5页

(54) 发明名称

一种并行的数据处理方法、装置和并行的数
据处理系统

(57) 摘要

本申请实施例公开了一种并行的数据处理方
法、装置和一种并行的数据处理系统。其中, 所述
方法包括: 主设备从数据源处获知需要进行处理
的待处理数据, 为每个待处理数据创建一个任务;
主设备在接收到从设备发送的获取任务的请求消
息时, 为发送请求的从设备分配任务, 对从设备返
回的执行结果进行合并, 并, 动态记录每个任务的
执行状态, 所述执行状态包括未执行、执行中、已
执行和已合并; 主设备输出被合并任务的执行结
果。根据本申请实施例, 可以支持系统在资源不足
或者资源浪费的情况下快速调整集群规模。



1. 一种并行的数据处理方法,其特征在于,包括:

主设备从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务;

主设备在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务;对从设备返回的执行结果进行合并,并且,动态记录每个任务的执行状态,所述执行状态包括未执行、执行中、已执行和已合并;

主设备输出被合并任务的执行结果;

其中,所述主设备从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务具体包括:主设备从数据源处获取需要进行处理的待处理数据的标识列表,所述标识列表中维护了所有待处理数据的数据标识;主设备从所述标识列表中提取每个待处理数据的标识,当为每个待处理数据创建一个任务后,将提取的标识放入任务中。

2. 根据权利要求1所述的并行的数据处理方法,其特征在于,所述方法还包括:

主设备在处于执行中状态的任务中,检查是否存在预置时间内未返回执行结果的任务,如果存在,将所述预置时间内未返回执行结果的任务重新标记为未执行。

3. 根据权利要求1或2所述的并行的数据处理方法,其特征在于,所述主设备动态记录每个任务的执行状态包括:

当主设备创建一个任务后,将创建的任务标记为未执行;并且,

当主设备接收到从设备返回的执行结果后,将执行完成的任务标记为已执行;并且,

当主设备检查到有处于已执行状态的任务,且对执行结果进行合并后,将被合并的任务标记为已合并。

4. 根据权利要求2所述的并行的数据处理方法,其特征在于,所述主设备在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务具体包括:

将新创建且处于未执行状态的任务优先分配给所述发送请求的从设备;

当新创建且处于未执行状态的任务被分配完后,再将重新标记为未执行状态的任务按照第一次被分配的时间顺序依次分配给所述发送请求的从设备。

5. 一种并行的数据处理装置,其特征在于,包括:

任务创建模块,用于从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务;

任务分配模块,用于在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务;

合并模块,用于对从设备返回的执行结果进行合并;

动态记录模块,用于动态记录每个任务的执行状态,所述执行状态包括未执行、执行中、已执行和已合并;

结果输出模块,用于输出被合并任务的执行结果;

其中,所述任务创建模块包括:

列表获取子模块,用于从数据源处获取需要进行处理的待处理数据的标识列表,所述标识列表中维护了所有待处理数据的数据标识;

标识提取子模块,用于从所述标识列表中提取每个待处理数据的标识,当为每个待处理数据创建一个任务后,将提取的标识放入任务中。

6. 根据权利要求 5 所述的并行的数据处理装置,其特征在于,还包括:

重记录模块,用于在处于执行中状态的任务中,检查是否存在预置时间内未返回执行结果的任务,如果存在,将所述预置时间内未返回执行结果的任务重新标记为未执行。

7. 一种并行的数据处理系统,其特征在于,包括:一个主设备和多个从设备,其中,

所述主设备,用于从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务;在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务;对从设备返回的执行结果进行合并,并动态记录每个任务的执行状态,所述执行状态包括未执行、执行中、已执行和已合并,输出被合并任务的合并结果;

所述从设备,用于向所述主设备发送获取任务的请求消息,当接收到所述主设备分配的任务后,执行分配的任务,将执行结果返回给所述主设备;

其中,所述主设备从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务具体包括:主设备从数据源处获取需要进行处理的待处理数据的标识列表,所述标识列表中维护了所有待处理数据的数据标识;主设备从所述标识列表中提取每个待处理数据的标识,当为每个待处理数据创建一个任务后,将提取的标识放入任务中。

8. 根据权利要求 7 所述的并行的数据处理系统,其特征在于,所述主设备还用于在状态为执行中的任务中,检查是否存在预置时间内未返回执行结果的任务,如果存在,将所述预置时间内未返回执行结果的任务重新标记为未执行。

一种并行的数据处理方法、装置和并行的数据处理系统

技术领域

[0001] 本申请涉及通信和计算机技术领域，特别是涉及一种并行的数据处理方法、装置和并行的数据处理系统。

背景技术

[0002] 随着 web2.0 技术的发展，互联网应用或者互联网平台中的业务数据，如用户行为数据和平台系统数据，都呈现出海量增长的趋势。为了适应对海量业务数据进行数据处理的应用需求，如，在互联网网站平台中，需要对用户行为数据和平台系统数据进行分析和计算，一种分布式的并行数据处理技术应运而生，其利用多个计算机相互协作工作，共同完成对海量数据的处理。

[0003] 当前，在大型的互联网网站平台中，应用最为广泛的一种分布式并行计算框架是 Hadoop 系统框架。请参阅图 1，其为现有技术中 Hadoop 系统框架的结构示意图。如图 1 所示，系统中包括一台主设备 (Master) 和一个从设备 (Slave) 集群，其中，每台从设备都具有逻辑功能上的数据节点 (DataNode) 和子任务追踪器 (TaskTracker)。DataNode 负责存储业务数据，TaskTracker 负责执行主设备推送的任务，即，对 DataNode 中存储的业务数据进行处理，并将任务的执行结果进行局部合并。主设备从逻辑功能来看，包括名字节点 (NameNode) 和任务追踪器 (JobTracker)。NameNode 负责管理各个从设备中存储的业务数据，JobTracker 负责启动、跟踪和调度各个从设备。

[0004] 但是，发明人在研究中发现，在现有的 Hadoop 系统中，主设备通过维护一个节点信息列表来管理集群中所有从设备的信息，并基于节点信息列表中的所有从设备信息制定了任务分配算法，按照任务分配算法将任务推送给各个从设备。然而，当系统中出现资源不足，需要动态扩展从设备时，或者出现资源浪费，需要删除从设备时，主设备必须先更新自身维护的节点信息列表，再基于更新的节点信息列表制定新的任务分配算法，以便主设备按照任务分配算法将任务推送给各个从设备，由从设备并行地进行数据处理。

[0005] 由此可见，现有 Hadoop 系统中并行的数据处理方法和相应的并行数据处理系统在扩展或者删除从设备时过程繁琐，并不利于动态扩展或者删除从设备，在资源不足或者资源浪费的情况下无法快速调整集群规模。

发明内容

[0006] 为了解决上述技术问题，本申请实施例提供了一种并行的数据处理方法和并行的数据处理系统，以支持系统在资源不足或者资源浪费的情况下快速调整集群规模。

[0007] 本申请实施例公开公开了如下技术方案：

[0008] 一种并行的数据处理方法，包括：主设备从数据源处获知需要进行处理的待处理数据，为每个待处理数据创建一个任务；主设备在接收到从设备发送的获取任务的请求消息时，为发送请求的从设备分配任务，对从设备返回的执行结果进行合并，并且，动态记录每个任务的执行状态，所述执行状态包括未执行、执行中、已执行和已合并；主设备输出被

合并任务的执行结果。

[0009] 一种并行的数据处理装置,包括:任务创建模块,用于从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务;任务分配模块,用于在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务;合并模块,用于对从设备返回的执行结果进行合并;动态记录模块,用于动态记录每个任务的执行状态,所述执行状态包括未执行、执行中、已执行和已合并;结果输出模块,用于输出被合并任务的执行结果。

[0010] 由上述实施例可以看出,由于主设备不再推送任务给从设备,而是在接收到从设备发送的获取任务的请求消息时,为从设备分配任务,同时,由于主设备不再通过维护一个节点信息列表来管理集群中所有从设备的信息,而是为每个待处理数据创建一个任务,并动态记录每个任务的执行状态。因此,对于主设备来说,从设备可以随时加入集群并向主设备请求分配任务,或者随时退出集群,在资源不足或者资源浪费的情况下可以快速调整集群规模。

附图说明

[0011] 为了更清楚地说明本申请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,对于本领域普通技术人员而言,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0012] 图 1 为本申请一种数据计费方法的一个实施例的流程图;

[0013] 图 2 为本申请一种并行的数据处理方法的一个实施例的流程图;

[0014] 图 3 为本申请一种系统应用场景示意图;

[0015] 图 4 为本申请中任务的状态转换图;

[0016] 图 5 为本申请一种并行的数据处理的交互流程图;

[0017] 图 6 为本申请一种并行的数据处理装置的一个实施例的结构图;

[0018] 图 7 为本申请一种并行的数据处理装置的另一个实施例的结构图;

[0019] 图 8 为本申请任务模块的一个实施例的结构图;

[0020] 图 9 为本申请一种并行的数据处理系统的一个实施例的结构图。

具体实施方式

[0021] 为使本申请的上述目的、特征和优点能够更加明显易懂,下面结合附图对本申请实施例进行详细描述。

[0022] 实施例一

[0023] 请参阅图 2,其为本申请一种并行的数据处理方法的一个实施例的流程图,该方法包括以下步骤:

[0024] 步骤 201:主设备从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务;

[0025] 其中,所述主设备从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务,具体可包括:主设备从数据源处获取需要进行处理的待处理数据的标识列表,所述标识列表中维护了所有待处理数据的数据标识;主设备从所述标识列表中提取每个待处理数据的标识,当为每个待处理数据创建一个任务后,将提取的标识放入该任务

中。

[0026] 例如,请参阅图 3,其为本申请一种系统应用场景示意图。如图 3 所示,数据源处具有需要进行处理的待处理数据的标识列表,在标识列表中维护了所有待处理数据的数据标识,例如,可以将数据的地址信息作为数据标识。当主设备从数据源处获取了标识列表后,根据标识列表中的各个数据标识就可以获知哪些数据资源为需要进行处理的待处理数据。当主设备为待处理数据创建一个任务后,从标识列表中提取出该待处理数据的数据标识,并将提取出的标识放入该任务中。例如,当为待处理数据 A 创建一个任务 1 后,主设备从标识列表中提取出待处理数据 A 的数据标识,并将待处理数据 A 的数据标识放入任务 1 中。

[0027] 步骤 202:主设备在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务,对从设备返回的执行结果进行合并;并且,动态记录每个任务的执行状态,所述执行状态包括:未执行、执行中、已执行和已合并;

[0028] 例如,与现有的 hadoop 系统中的任务推送机制不同,本申请实施例中主设备向从设备分配任务是基于从设备的请求机制,即,当主设备收到从设备发送的获取任务的请求消息时,再为发送请求的从设备分配任务。当有从设备返回任务的执行结果时,对从设备返回的执行结果进行合并。同时,主设备动态记录每个任务的执行状态,所述执行状态包括:未执行、执行中、已执行和已合并。

[0029] 其中,所述主设备动态记录每个任务的执行状态具体为:当主设备创建一个任务后,将创建的任务标记为未执行;并且,当主设备接收到从设备返回的执行结果后,将执行完成的任务标记为已执行;并且,当主设备检查到有处于已执行状态的任务,并对执行结果进行合并后,将被合并的任务标记为已合并。

[0030] 需要说明的是,由于主设备创建一个任务的时间、接收到从设备返回的执行结果的时间和检查是否有处于已执行状态的周期时间并没有严格的先后顺序,因此,本申请实施例也并不限定上述三个标记过程的执行顺序。例如,每当主设备新创建一个任务时,由于新创建的任务没有分配给从设备,也没有被从设备执行,因此,将该新创建的任务标记为未执行。每当主设备接收到某个从设备返回的某个任务 A 的执行结果后,将执行完成的任务 A 标记为已执行。每当主设备检查是否有处于已执行状态任务的检查周期到达,且对执行结果进行合并后,将被合并的任务标记为已合并。

[0031] 还需要说明的是,主设备除了可以周期性检查是否有处于已执行状态的任务之外,主设备也可以每接收到从设备返回的执行结果,就检查一次是否有处于已执行状态的任务,本申请实施例对此并不限定。当然,前一种方法可以有效地节省系统功耗。

[0032] 步骤 203:主设备输出被合并任务的执行结果。

[0033] 需要说明的是,主设备可以预先输出部分被合并任务的结果,也可以是当所有任务都处于已合并状态时,主设备输出所有被合并任务的执行结果。例如,主设备周期性地查看所有任务的执行状态,当所有任务的执行状态都为已合并时,输出所有被合并任务的执行结果。

[0034] 请参阅图 4,其为本申请中任务的状态转换图。如图 4 所示,当任务被创建时,其状态为未执行;当有从设备请求获取任务,且主设备从处于未执行状态的任务中选择任务并分配给从设备时,任务的状态从未执行转换为执行中;当任务分配后的预置时间内,主设备没有收到从设备反馈的执行结果,任务的状态从执行中又转换为未执行;当从设备完成任

务并将执行结果反馈给主设备后,任务的状态从执行中转换为已执行;当主设备将已执行状态的任务进行合并后,任务的状态从已执行转换为已合并。

[0035] 在现有技术中,为了监测任务处理是否处于异常状况时,主设备需要向多个从设备反复轮询执行状况,因此,导致任务的执行效率较低,系统的稳定性和可用性也较低。为了进一步提高任务的执行效率、系统的稳定性和可用性,优选的,本申请实施例的方法还包括:主设备在处于执行中状态的任务中,周期性检查是否存在预置时间内未返回执行结果的任务,如果存在,将所述预置时间内未返回执行结果的任务重新标记为未执行。

[0036] 例如,主设备将任务 A 分配给某个从设备后,将任务 A 标记为执行中,同时设定一个定时器,该定时器的定时时间为一预置时间,当定时器超时,如果主设备仍然没有收到任务 A 的执行结果,则重新将任务 A 标记为未执行。

[0037] 此时,在主设备侧,除了包括一部分新创建且标记为未执行的任务之外,还包括一部分由于在预置时间内未被执行完而重新被标记为未执行状态的任务,当主设备在接收到从设备发送的获取任务的请求消息时,可以任意选择一个当前处于未执行状态的任务,并分配给发送请求的从设备。优选地,可以将新创建且处于未执行状态的任务优先分配给所述发送请求的从设备;当新创建且处于未执行状态的任务被分配完毕后,再将重新标记为未执行状态的任务按照曾被分配的时间顺序一次分配给所述发送请求的从设备。

[0038] 为了能够简单说明主设备分配任务的过程,以在主设备侧共有 5 个处于未执行状态的任务为例,其中的任务 1、任务 2 和任务 3 为新创建且处于未执行状态的任务,任务 4 和任务 5 为重新标记为未执行状态的任务,并且,任务 4 第一次被分配的时间早于任务 5 第一次被分配的时间。初始时,主设备优先将任务 1、任务 2 和任务 3 分配给从设备,当任务 1、任务 2 和任务 3 都被分配完毕后,主设备先将任务 4 分配给从设备,再将任务 5 分配给从设备。

[0039] 由上述实施例可以看出,由于主设备不再推送任务给从设备,而是在接收到从设备发送的获取任务的请求消息时,为从设备分配任务,同时,由于主设备不再通过维护一个节点信息列表来管理集群中所有从设备的信息,而是为每个待处理数据创建一个任务,并动态记录每个任务的执行状态。因此,对于主设备来说,从设备可以随时加入集群并向主设备请求分配任务,或者随时退出集群,在资源不足或者资源浪费的情况下可以快速调整集群规模。

[0040] 另外,由于从设备任务的执行不再由主设备全程监控,主设备仅仅维护任务状态,一旦任务被分配后的一定时间内没有返回,即认定任务的执行出现了异常情况,将任务状态重新标记为未执行,对任务重新进行分配。从而进一步提高了任务的执行效率、系统的稳定性和可用性。

[0041] 实施例二

[0042] 下面从主设备与从设备交互过程来详细说明并行的数据处理方法。请参阅图 5,其为本申请一种并行的数据处理的交互流程图,如图 5 所示,所述交互流程包括:

[0043] 步骤 501:主设备从数据源获取需要进行处理的待处理数据的标识列表;

[0044] 其中,数据源可以是 ftp 服务器、数据库 (DB) 或文件系统。通过标识列表,主设备可以获知哪些数据为待处理数据。

[0045] 步骤 502:主设备为标识列表中所标识的每个待处理数据创建一个任务,使用一

个任务队列维护所有任务,并将新创建的任务标记为未执行状态;

[0046] 其中,主设备在创建任务时,还将每个待处理数据的数据标识放入相对应的任务中。

[0047] 步骤 503:主设备接收从设备发送的获取任务的请求消息;

[0048] 步骤 504:从任务队列中将处于未执行状态的任务分配给发送请求的从设备,并将任务的状态从未执行标记为执行中;

[0049] 步骤 505:从设备接收到主设备发送的任务后,从任务中解析得到待处理数据的数据标识;

[0050] 步骤 506:从设备根据数据标识,从数据源中获取待处理数据;

[0051] 步骤 507:从设备对获取的待处理数据进行分析 and 计算;

[0052] 以上步骤 505-507 为从设备执行任务的过程,其中,对待处理数据的分析和计算方法可以采用与现有技术相同的方法,故本申请实施例对此不再赘述。

[0053] 步骤 508:从设备将计算和分析的结果返回给主设备,并向主设备发送获取下一个任务的请求消息;

[0054] 步骤 509:主设备接收到从设备返回的执行结果后,将任务的状态从执行中标记为已执行;

[0055] 步骤 510:主设备检查任务队列中是否有处于已执行状态的任务,如果有,对任务的执行结果进行合并,将任务的状态从已执行标记为已合并,如果否,等待下一次检查;

[0056] 其中,主设备对已执行状态的任务的检查可以是周期性的,也可以是由从设备返回执行结果时触发下一次检查。

[0057] 步骤 511:主设备检查任务队列中是否所有任务都处于已合并状态,如果是,输出所有被合并任务的执行结果,如果否,等待下一次检查;

[0058] 其中,主设备对已合并状态的任务的检查可以是周期性的。

[0059] 步骤 512:主设备在处于执行中状态的任务中,检查是否存在预置时间内未返回执行结果的任务,如果存在,将所述预置时间内未返回执行结果的任务重新标记为未执行。

[0060] 需要说明的是,步骤 510- 步骤 512 与其他步骤 501-509 没有严格的执行先后顺序,并且,步骤 510- 步骤 512 之间也没有严格的执行先后顺序,当其到达下一次检查时,即可执行该步骤。

[0061] 由上述实施例可以看出,由于主设备不再推送任务给从设备,而是在接收到从设备发送的获取任务的请求消息时,为从设备分配任务,同时,由于主设备不再通过维护一个节点信息列表来管理集群中所有从设备的信息,而是为每个待处理数据创建一个任务,并动态记录每个任务的执行状态。因此,对于主设备来说,从设备可以随时加入集群并向主设备请求分配任务,或者随时退出集群,在资源不足或者资源浪费的情况下可以快速调整集群规模。

[0062] 另外,由于从设备任务的执行不再由主设备全程监控,主设备仅仅维护任务状态,一旦任务被分配后的一定时间内没有返回,即认定任务的执行出现了异常情况,将任务状态重新标记为未执行,对任务重新进行分配。从而进一步提高了任务的执行效率、系统的稳定性和可用性。

[0063] 实施例三

[0064] 与上述一种并行的数据处理方法相对应,本申请实施例还提供了一种并行的数据处理装置。请参阅图 6,其为本申请一种并行的数据处理装置的一个实施例的结构图,该装置包括任务创建模块 601、任务分配模块 602、合并模块 603、动态记录模块 604 和结果输出模块 605。下面结合该装置的工作原理进一步介绍其内部结构以及连接关系。

[0065] 任务创建模块 601,用于从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务;

[0066] 任务分配模块 602,用于在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务;

[0067] 合并模块 603,用于对从设备返回的执行结果进行合并;

[0068] 动态记录模块 604,用于动态记录每个任务的执行状态,所述执行状态包括未执行、执行中、已执行和已合并;

[0069] 结果输出模块 605,用于输出被合并任务的执行结果。

[0070] 优选的,请参阅图 7,其为本申请一种并行的数据处理装置的另一个实施例的结构图,如图 7 所示,所述装置还包括:重记录模块 606,用于在处于执行中状态的任务中,检查是否存在预置时间内未返回执行结果的任务,如果存在,将所述预置时间内未返回执行结果的任务重新标记为未执行。

[0071] 优选的,请参阅图 8,其为本申请任务创建模块的一个实施例的结构图,任务创建模块包括:列表获取子模块 801 和标识提取子模块 802,其中,

[0072] 列表获取子模块 801,用于从数据源处获取需要进行处理的待处理数据的标识列表,所述标识列表中维护了所有待处理数据的数据标识;

[0073] 标识提取子模块 802,用于从所述标识列表中提取每个待处理数据的标识,当为每个待处理数据创建一个任务后,将提取的标识放入任务中。

[0074] 优选的,动态记录模块包括:第一标记子模块,当创建一个任务后,将创建的任务标记为未执行;第二标记子模块,用于当接收到从设备返回的执行结果后,将执行完成的任务标记为已执行;第三标记子模块,用于当周期性检查到有处于已执行状态的任务,且对执行结果进行合并后,将被合并的任务标记为已合并。

[0075] 针对图 7 中的并行的数据处理装置,优选地,任务分配模块包括:第一分配子模块,用于将新创建且处于未执行状态的任务优先分配给所述发送请求的从设备;第二分配子模块,用于当新创建且处于未执行状态的任务被分配完后,再将重新标记为未执行状态的任务按照第一次被分配的时间顺序依次分配给所述发送请求的从设备。

[0076] 由上述实施例可以看出,由于主设备不再推送任务给从设备,而是在接收到从设备发送的获取任务的请求消息时,为从设备分配任务,同时,由于主设备不再通过维护一个节点信息列表来管理集群中所有从设备的信息,而是为每个待处理数据创建一个任务,并动态记录每个任务的执行状态。因此,对于主设备来说,从设备可以随时加入集群并向主设备请求分配任务,或者随时退出集群,在资源不足或者资源浪费的情况下可以快速调整集群规模。

[0077] 另外,由于从设备任务的执行不再由主设备全程监控,主设备仅仅维护任务状态,一旦任务被分配后的一定时间内没有返回,即认定任务的执行出现了异常情况,将任务状态重新标记为未执行,对任务重新进行分配。从而进一步提高了任务的执行效率、系统的稳

定性和可用性。

[0078] 实施例四

[0079] 本申请实施例还提供了一种并行的数据处理系统。请参阅图9,其为本申请一种并行的数据处理系统的一个实施例的结构图,该系统包括:一个主设备901和多个从设备902组成的集群。下面结合该装置的工作原理进一步介绍其内部结构以及连接关系。

[0080] 主设备901,用于从数据源处获知需要进行处理的待处理数据,为每个待处理数据创建一个任务,在接收到从设备发送的获取任务的请求消息时,为发送请求的从设备分配任务,对从设备返回的执行结果进行合并,并动态记录每个任务的执行状态,所述执行状态包括未执行、执行中、已执行和已合并,输出被合并任务的合并结果;

[0081] 从设备902,用于向所述主设备发送获取任务的请求消息,当接收到所述主设备分配的任务后,执行分配的任务,将执行结果返回给所述主设备。

[0082] 优选的,主设备901还用于在状态为执行中的任务中,检查是否存预置时间内未返回执行结果的任务,如果存在,将所述预置时间内未返回执行结果的任务重新标记为未执行。

[0083] 由上述实施例可以看出,由于主设备不再推送任务给从设备,而是在接收到从设备发送的获取任务的请求消息时,为从设备分配任务,同时,由于主设备不再通过维护一个节点信息列表来管理集群中所有从设备的信息,而是为每个待处理数据创建一个任务,并动态记录每个任务的执行状态。因此,对于主设备来说,从设备可以随时加入集群并向主设备请求分配任务,或者随时退出集群,在资源不足或者资源浪费的情况下可以快速调整集群规模。

[0084] 另外,由于从设备任务的执行不再由主设备全程监控,主设备仅仅维护任务状态,一旦任务被分配后的一定时间内没有返回,即认定任务的执行出现了异常情况,将任务状态重新标记为未执行,对任务重新进行分配。从而进一步提高了任务的执行效率、系统的稳定性和可用性。

[0085] 需要说明的是,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,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory, ROM)或随机存储记忆体(Random Access Memory, RAM)等。

[0086] 以上对本申请所提供的一种并行的数据处理方法、装置和并行的数据处理系统进行了详细介绍,本文中应用了具体实施例对本申请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申请的方法及其核心思想;同时,对于本领域的一般技术人员,依据本申请的思想,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申请的限制。

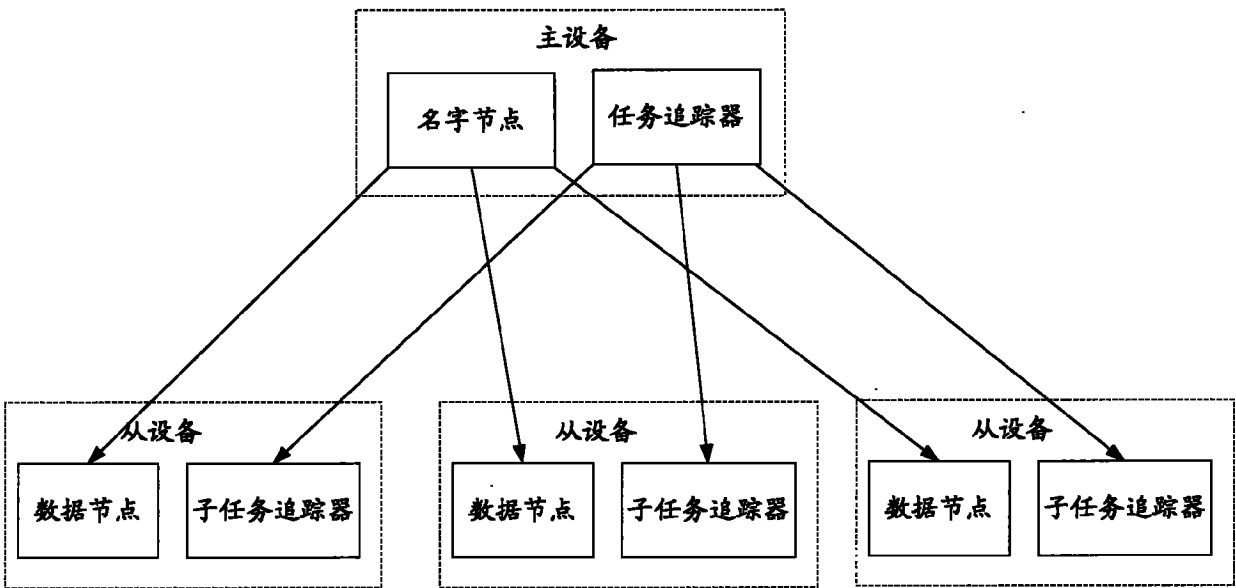


图 1

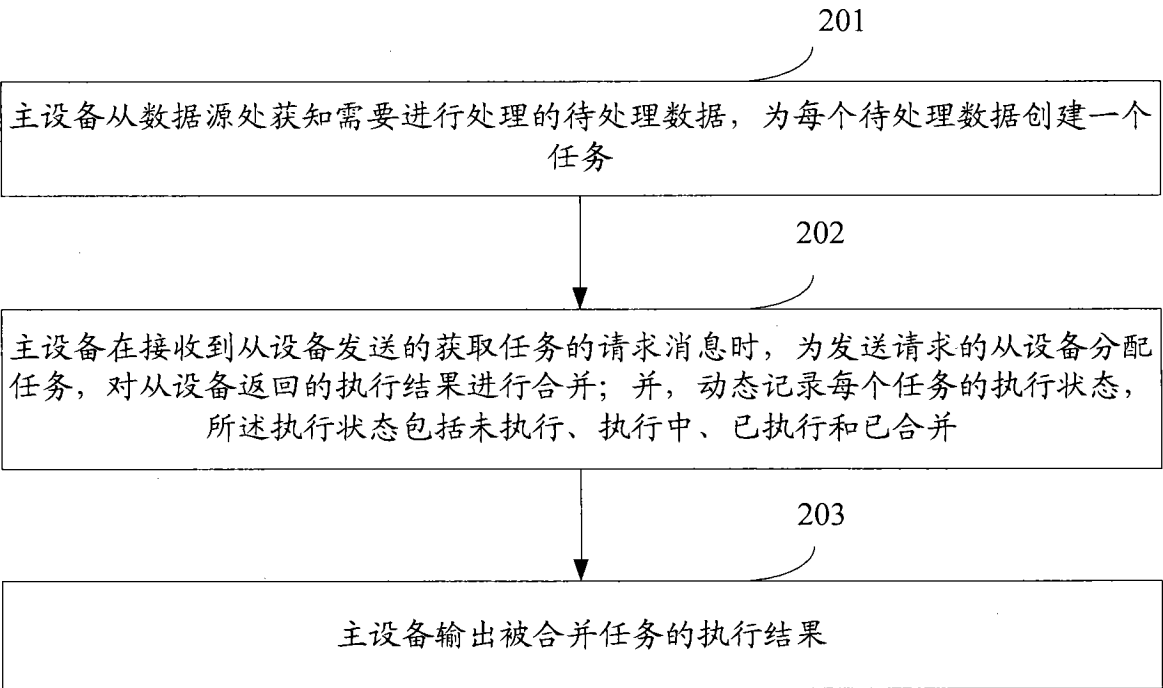


图 2

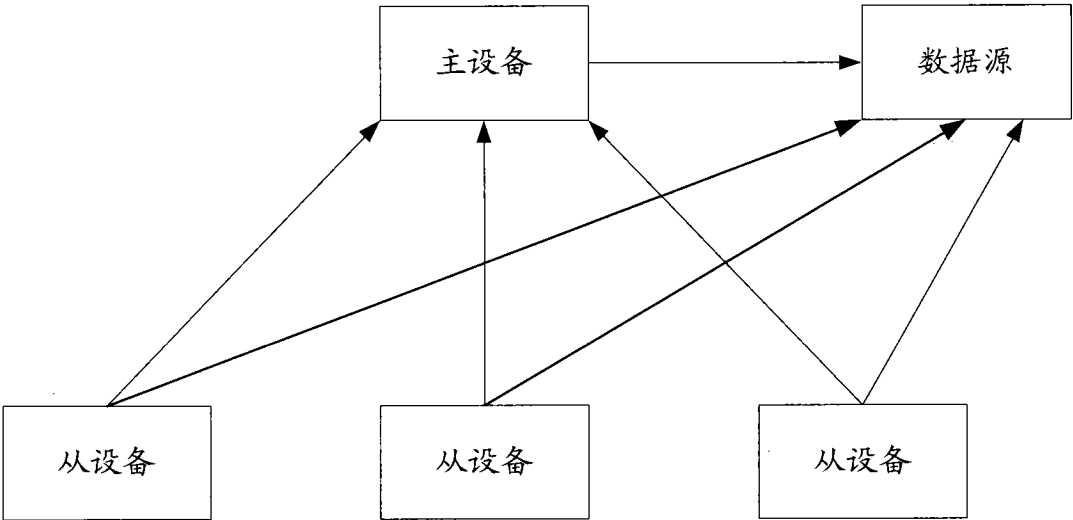


图 3

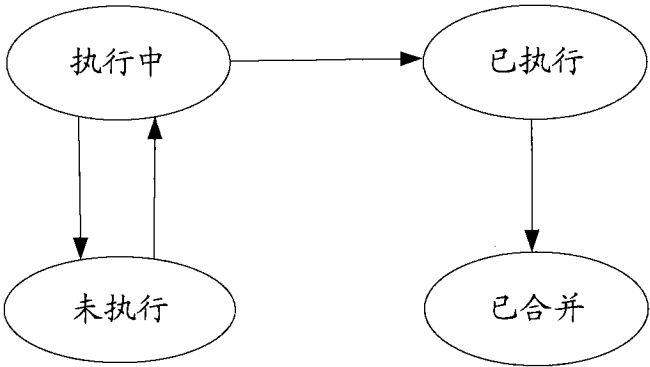


图 4

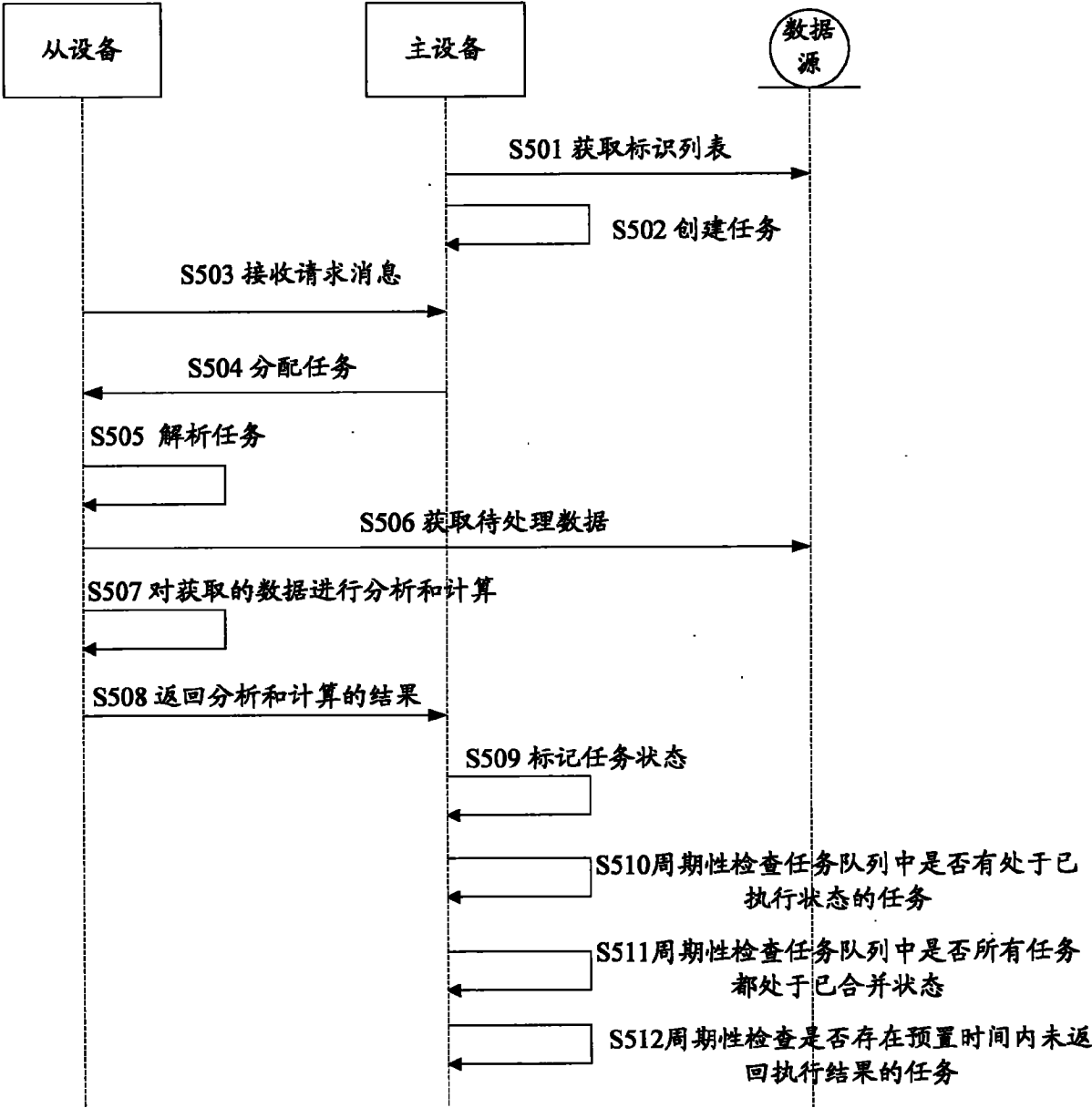


图 5

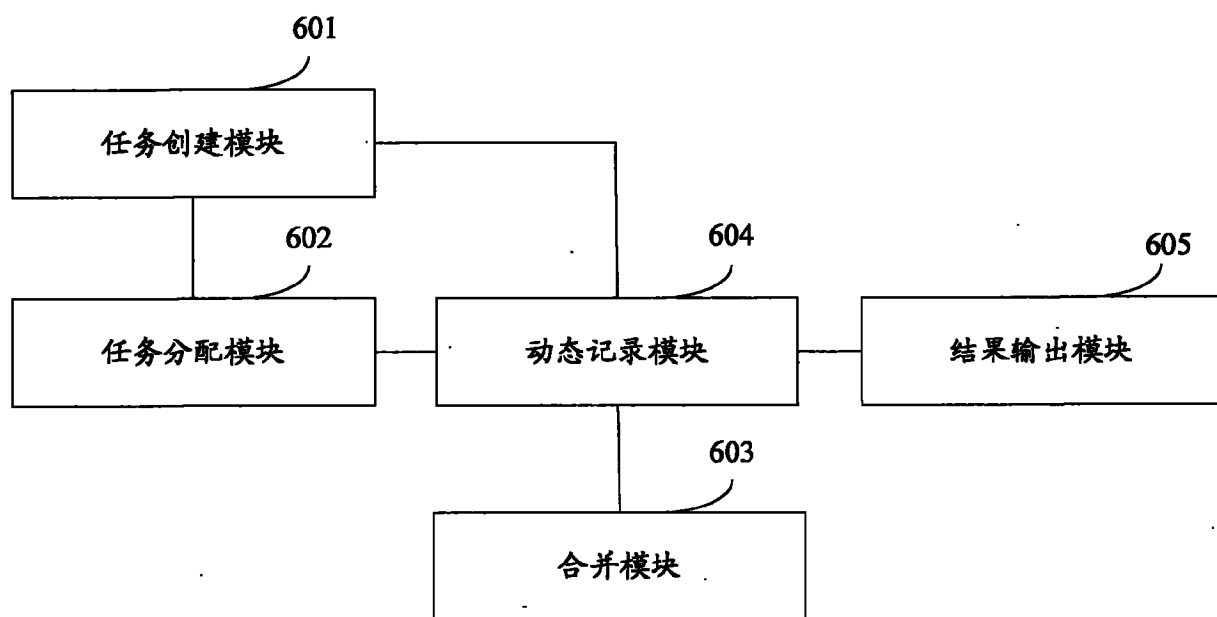


图 6

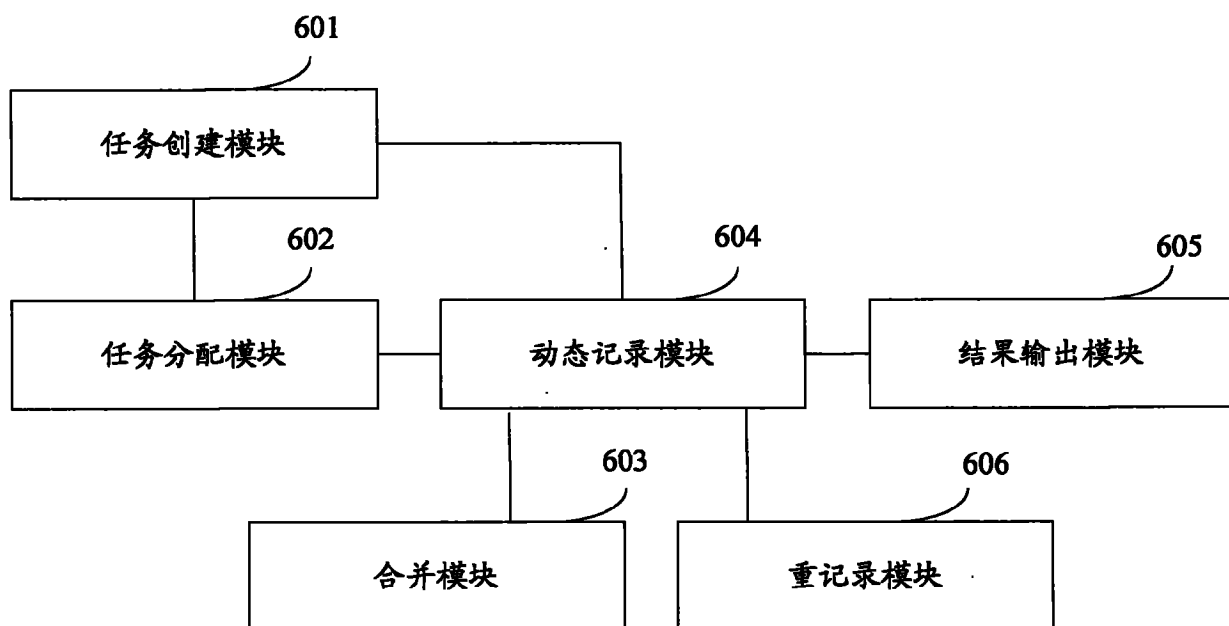


图 7

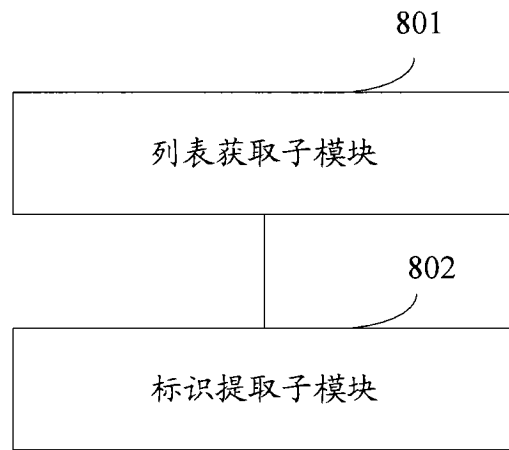


图 8

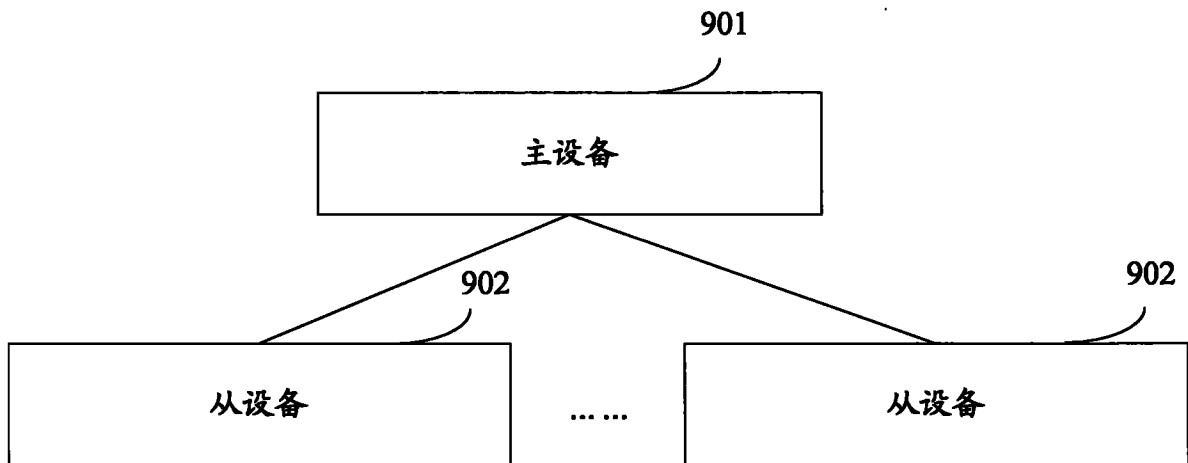


图 9