(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115604010 A
(43)申请公布日 2023.01.13

(21)申请号 202211275148.7

(22)申请日 2022.10.18

(71)申请人 北京富算科技有限公司
地址 100020 北京市朝阳区东三环中路9号19层2201

(72)发明人 朱崇炳 卞阳 赵东 尤志强

(74)专利代理机构 上海弼兴律师事务所 31283
专利代理师 罗朗 林嵩

(51)Int.Cl.
H04L 9/40(2022.01)

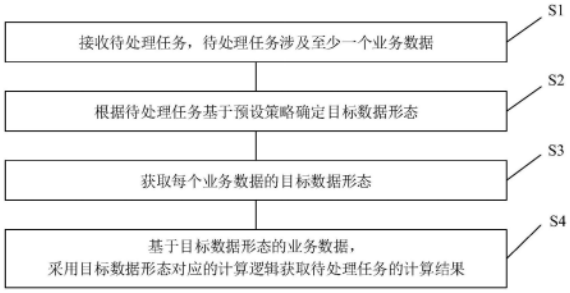
权利要求书2页 说明书11页 附图6页

(54)发明名称

数据混合处理方法、系统、设备、介质及数据处理系统

(57)摘要

本公开提供了一种数据混合处理方法、系统、设备、介质及数据处理系统。该数据处理方法应用于部署有多种数据形态对应的计算逻辑的数据处理节点，该数据混合处理方法包括：接收待处理任务，待处理任务涉及至少一个业务数据；根据待处理任务基于预设策略确定目标数据形态；获取每个业务数据的目标数据形态；基于目标数据形态的业务数据，采用目标数据形态对应的计算逻辑获取待处理任务的计算结果。该方法基于部署多种数据形态对应的计算逻辑的数据处理节点，可以进行不同数据形态的业务数据计算。不再需要将不同数据形态的计算逻辑分开管理和审计，有利于数据混合处理系统的开发和管理。



CN 115604010 A

1.一种数据混合处理方法,其特征在于,所述数据混合处理方法应用于数据处理节点,所述数据处理节点部署有多种数据形态对应的计算逻辑,所述数据混合处理方法包括:

接收待处理任务,所述待处理任务涉及至少一个业务数据;

根据所述待处理任务基于预设策略确定目标数据形态;

获取每个所述业务数据的所述目标数据形态;

基于所述目标数据形态的业务数据,采用所述目标数据形态对应的所述计算逻辑获取所述待处理任务的计算结果。

2.根据权利要求1所述的数据混合处理方法,其特征在于,所述根据所述待处理任务基于预设策略确定目标数据形态的步骤包括:

获取所述待处理任务涉及的每个所述业务数据的初始数据形态,并根据每个所述业务数据的所述初始数据形态对应的安全级别确定所述目标数据形态;

和/或,

所述根据所述待处理任务基于预设策略确定目标数据形态的步骤包括:

当所述待处理任务涉及多个所述业务数据时,判断所述业务数据是否来自于多个用户,若是,按照所述用户之间的约定数据形态确定所述目标数据形态。

3.根据权利要求1所述的数据混合处理方法,其特征在于,所述获取每个所述业务数据的所述目标数据形态的步骤包括:

基于每个所述业务数据的数据封装对象,调用对应所述目标数据形态的获取方法获取每个所述业务数据的所述目标数据形态;

其中,所述数据封装对象包括所述业务数据的初始数据形态,以及所述业务数据不同数据形态的获取方法,以及用于将所述初始数据形态转换成其他数据形态的预设转换算法。

4.根据权利要求1所述的数据混合处理方法,其特征在于,所述待处理任务为原始计算任务拆分成的子任务,不同的所述数据处理节点接收不同的所述子任务,接收不同所述子任务的所述数据处理节点串行或并行进行处理。

5.根据权利要求1-4中任一项所述的数据混合处理方法,其特征在于,所述数据形态包括明文形态和密文形态。

6.根据权利要求5所述的数据混合处理方法,其特征在于,所述根据所述待处理任务基于预设策略确定目标数据形态的步骤包括:

获取所述待处理任务涉及的每个所述业务数据的初始数据形态;

判断每个所述业务数据的所述初始数据形态是否为所述密文形态;

若任一所述业务数据的初始数据形态为所述密文形态,所述目标数据形态为所述密文形态,否则所述目标数据形态为所述明文形态。

7.一种数据混合处理系统,其特征在于,所述数据混合处理系统部署于数据处理节点,所述数据处理节点还部署有多种数据形态对应的计算逻辑,所述数据混合处理系统包括:

任务接收模块,用于接收待处理任务,所述待处理任务涉及至少一个业务数据;

数据形态确定模块,用于根据所述待处理任务基于预设策略确定目标数据形态;

数据获取模块,用于获取每个所述业务数据的所述目标数据形态;

任务处理模块,用于基于所述目标数据形态的业务数据,采用所述目标数据形态对应

的所述计算逻辑获取所述待处理任务的计算结果。

8.一种数据处理系统，其特征在于，所述数据处理系统包括控制端和若干数据处理节点，所述控制端用于为所述数据处理节点发送待处理任务，所述数据处理节点部署有多种数据形态对应的计算逻辑以及权利要求7所述的数据混合处理系统。

9.一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，其特征在于，所述处理器执行计算机程序时，实现如权利要求1-6中任一项所述的数据混合处理方法。

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如权利要求1-6中任一项所述的数据混合处理方法。

数据混合处理方法、系统、设备、介质及数据处理系统

技术领域

[0001] 本公开涉及数据处理技术领域，特别涉及一种数据混合处理方法、系统、设备、介质及数据处理系统。

背景技术

[0002] 传统的数据混合处理系统，包括控制端以及用于处理不同数据形态的数据的若干个处理节点。该系统在单个用户端使用时，控制端需要先将计算任务拆分成多个待处理任务，判断哪些待处理任务是可以并发同步进行的，以及判断哪些待处理任务是需要等待其他待处理任务完成后才能进行的，然后根据待处理任务中涉及到的业务数据的数据形态，将其分发给相应的处理节点进行处理。

[0003] 该系统在多个用户端使用时，涉及到两个用户端之间的数据交互以及联合运算时，由于不能直接将用户端内的业务数据进行传输，控制端还需要将每个用户端的发出方从自身使用数据的数据形态转换成与接收方约定的数据形态，以保证业务数据的数据安全。

[0004] 但在传统的数据混合处理系统中，不同数据形态的数据的计算逻辑需要分开管理和审计，不利于系统的开发和管理。特别是，一个待处理任务中涉及到多种数据形态的数据混合处理时，控制端不仅需要明确区分待处理任务使用数据的数据形态，还需要显示地调用数据形态之间的转换方法，增加了控制端的开发难度。

[0005] 另外，在部署系统时，还要明确区分用于处理不同数据形态的处理节点，导致部署过程复杂。并且，由于用于处理不同数据形态的处理节点是事先部署好的，也就是说，处理每一种数据形态的处理节点资源是有限的。在某次计算任务采用大量的某一种数据形态进行计算时，会导致用于处理其他数据形态的处理节点大量闲置，而待处理任务不能及时得到处理造成系统拥堵，最终可能无法在规定时间内完成计算任务，导致计算任务失败。

发明内容

[0006] 为了解决上述技术问题，本公开提供一种数据混合处理方法、系统、设备、介质及数据处理系统，通过在数据处理节点部署多种数据形态对应的计算逻辑，使得数据处理节点可以进行不同数据形态的业务数据计算。不再需要将不同数据形态的计算逻辑分开管理和审计，有利于数据混合处理系统的开发和管理。

[0007] 第一方面，本公开提供一种数据混合处理方法，该方法应用于数据处理节点，所述数据处理节点部署有多种数据形态对应的计算逻辑。

[0008] 具体地，该数据混合处理方法包括：

[0009] 接收待处理任务，所述待处理任务涉及至少一个业务数据；

[0010] 根据所述待处理任务基于预设策略确定目标数据形态；

[0011] 获取每个所述业务数据的所述目标数据形态；

[0012] 基于所述目标数据形态的业务数据，采用所述目标数据形态对应的所述计算逻辑

获取所述待处理任务的计算结果。

[0013] 较佳地，所述根据所述待处理任务基于预设策略确定目标数据形态的步骤包括：

[0014] 获取所述待处理任务涉及的每个所述业务数据的初始数据形态，并根据每个所述业务数据的所述初始数据形态对应的安全级别确定所述目标数据形态。

[0015] 较佳地，所述根据所述待处理任务基于预设策略确定目标数据形态的步骤包括：

[0016] 当所述待处理任务涉及多个所述业务数据时，判断所述业务数据是否来自于多个用户，若是，按照所述用户之间的约定数据形态确定所述目标数据形态。

[0017] 较佳地，所述获取每个所述业务数据的所述目标数据形态的步骤包括：

[0018] 基于每个所述业务数据的数据封装对象，调用对应所述目标数据形态的获取方法获取每个所述业务数据的所述目标数据形态；

[0019] 其中，所述数据封装对象包括所述业务数据的初始数据形态，以及所述业务数据不同数据形态的获取方法，以及用于将所述初始数据形态转换成其他数据形态的预设转换算法。

[0020] 较佳地，所述待处理任务为原始计算任务拆分成的子任务，不同的所述数据处理节点接收不同的所述子任务，接收不同所述子任务的所述数据处理节点串行或并行进行处理。

[0021] 较佳地，所述数据形态包括明文形态和密文形态。

[0022] 较佳地，所述根据所述待处理任务基于预设策略确定目标数据形态的步骤包括：

[0023] 获取所述待处理任务涉及的每个所述业务数据的初始数据形态；

[0024] 判断每个所述业务数据的所述初始数据形态是否为所述密文形态；

[0025] 若任一所述业务数据的初始数据形态为所述密文形态，所述目标数据形态为所述密文形态，否则所述目标数据形态为所述明文形态。

[0026] 第二方面，本公开提供一种数据混合处理系统，该系统部署于数据处理节点，所述数据处理节点还部署有多种数据形态对应的计算逻辑。

[0027] 具体地，该数据混合处理系统包括：

[0028] 任务接收模块，用于接收待处理任务，所述待处理任务涉及至少一个业务数据；

[0029] 数据形态确定模块，用于根据所述待处理任务基于预设策略确定目标数据形态；

[0030] 数据获取模块，用于获取每个所述业务数据的所述目标数据形态；

[0031] 任务处理模块，用于基于所述目标数据形态的业务数据，采用所述目标数据形态对应的所述计算逻辑获取所述待处理任务的计算结果。

[0032] 较佳地，所述数据形态确定模块具体用于：获取所述待处理任务涉及的每个所述业务数据的初始数据形态，并根据每个所述业务数据的所述初始数据形态对应的安全级别确定所述目标数据形态。

[0033] 较佳地，所述数据形态确定模块具体还用于：当所述待处理任务涉及多个所述业务数据时，判断所述业务数据是否来自于多个用户，若是，按照所述用户之间的约定数据形态确定所述目标数据形态。

[0034] 较佳地，所述数据获取模块具体用于：基于每个所述业务数据的数据封装对象，调用对应所述目标数据形态的获取方法获取每个所述业务数据的所述目标数据形态；其中，所述数据封装对象包括所述业务数据的初始数据形态，以及所述业务数据不同数据形态的

获取方法，以及用于将所述初始数据形态转换成其他数据形态的预设转换算法。

[0035] 较佳地，所述待处理任务为原始计算任务拆分成的子任务，不同的所述数据处理节点接收不同的所述子任务，接收不同所述子任务的所述数据处理节点串行或并行进行处理。

[0036] 较佳地，所述数据形态包括明文形态和密文形态。

[0037] 较佳地，所述根据所述待处理任务基于预设策略确定目标数据形态的步骤包括：

[0038] 获取所述待处理任务涉及的每个所述业务数据的初始数据形态；

[0039] 判断每个所述业务数据的所述初始数据形态是否为所述密文形态；

[0040] 若任一所述业务数据的初始数据形态为所述密文形态，所述目标数据形态为所述密文形态，否则所述目标数据形态为所述明文形态。

[0041] 第三方面，本公开提供一种数据处理系统，包括控制端和若干数据处理节点，所述控制端用于为所述数据处理节点发送待处理任务，所述数据处理节点部署有多种数据形态对应的计算逻辑，以及第一方面及其任一种实施方式所述的数据混合处理系统。

[0042] 第四方面，本公开提供一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行计算机程序时，实现第一方面及其任一种实施方式所述的数据混合处理方法。

[0043] 第五方面，本公开提供一种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，实现第一方面及其任一种实施方式所述的数据混合处理方法。

[0044] 本公开的积极进步效果在于：

[0045] 本公开提供了一种数据混合处理方法，该方法基于部署多种数据形态对应的计算逻辑的数据处理节点，可以进行不同数据形态的业务数据计算。不再需要将不同数据形态的计算逻辑分开管理和审计，有利于数据混合处理系统的开发和管理。

[0046] 基于该数据混合处理方法，本公开还提出了一种数据混合处理系统，将该数据混合处理系统部署于数据处理系统的数据处理节点上，不再需要将不同数据形态的计算逻辑分开管理和审计，也有利于数据处理系统的开发和管理。并且，极大程度地提高了数据处理系统的并发能力，提高原始计算任务的子任务的处理速度，进而提升原始计算任务的成功率。另外，在部署数据处理系统时，不再需要部署用于处理不同数据形态的数据处理节点，简化系统部署过程。

[0047] 特别是，一个待处理任务中涉及到多种数据形态的数据混合处理时，控制端不再需要明确区分待处理任务使用数据的数据形态，也不再需要显示地调用数据形态之间的转换方法，极大程度地减少了控制端的开发难度。

附图说明

[0048] 图1为本公开实施例1中的数据混合处理方法的流程示意图；

[0049] 图2为本公开实施例1中的确定目标数据形态的流程示意图；

[0050] 图3为本公开实施例1中的数据封装类的结构示意图；

[0051] 图4为本公开实施例2中的数据混合处理系统的模块示意图；

[0052] 图5为本公开实施例3中的数据处理系统的系统架构图；

[0053] 图6为本公开实施例3中传统的用于明密文数据混合处理的数据处理系统的系统

架构图；

[0054] 图7为本公开实施例3中用于明密文数据混合处理的数据处理系统优化后的系统架构图；

[0055] 图8为本公开实施例3中数据处理系统在实际应用场景中的系统架构图；

[0056] 图9为本公开实施例4提供的电子设备的结构示意图。

具体实施方式

[0057] 下面通过实施例的方式进一步说明本公开，但并不因此将本公开限制在所述的实施例范围之中。

[0058] 实施例1

[0059] 本实施例提供一种数据混合处理方法，数据混合处理方法应用于数据处理节点，数据处理节点部署有多种数据形态对应的计算逻辑。

[0060] 如图1所示，该数据混合处理方法包括步骤S1-S4：

[0061] 步骤S1、接收待处理任务，待处理任务涉及至少一个业务数据；

[0062] 步骤S2、根据待处理任务基于预设策略确定目标数据形态；

[0063] 步骤S3、获取每个业务数据的目标数据形态；

[0064] 步骤S4、基于目标数据形态的业务数据，采用目标数据形态对应的计算逻辑获取待处理任务的计算结果。

[0065] 在步骤S1中，数据处理节点接收待处理任务，该待处理任务通常为对一个业务数据进行运算或对若干个业务数据进行同级运算。因此，一个待处理任务涉及至少一个业务数据。

[0066] 在一种可行的实施方式中，待处理任务为原始计算任务拆分成的子任务，不同的数据处理节点会接收到不同的子任务，接收不同子任务的数据处理节点串行或并行进行处理。

[0067] 例如，如果多个数据处理节点接收到的待处理任务所涉及的业务数据可以直接获取得到，那么这些数据处理节点可以并行处理对应的待处理任务。如果某些待处理任务所涉及的业务数据需要等待其他数据处理节点返回的计算结果，这些待处理任务会需要的计算结果返回后才被分发至数据处理节点，因此，接收此类待处理任务的数据处理节点相对于为其返回计算结果的数据处理节点来说，是串行进行处理的。

[0068] 在步骤S2中，根据预设策略确定数据处理节点需要在数据处理的过程中使用的目标数据形态。

[0069] 在一种可行的实施方式中，步骤S2包括：获取待处理任务涉及的每个业务数据的初始数据形态，并根据每个业务数据的初始数据形态对应的安全级别确定目标数据形态。

[0070] 在另一种可行的实施方式中，步骤S2包括：当待处理任务涉及多个业务数据时，判断业务数据是否来自于多个用户，若是，按照用户之间的约定数据形态确定目标数据形态。

[0071] 也就是说，预设策略主要围绕着数据安全进行设置。一种策略是，在待处理任务涉及对应不同安全级别的多个业务数据时，通常选用这些业务数据中最高安全级别对应的数据形态作为目标数据形态。另一种策略是，在待处理任务涉及多个业务数据的基础上，进一步判断业务数据是否来自不同的用户，如果是，为了防止业务数据在传输或者处理的过程

中泄露，将用户之间约定好的数据形态作为目标数据形态。此时的目标数据形态通常是需要加密的。

[0072] 在实际应用场景中，待处理任务涉及第一业务数据和第二业务数据，第一业务数据的初始数据形态为明文形态，第二业务数据的初始数据形态为密文形态。显然地，密文形态的安全级别高于明文形态。因此，目标数据形态应为密文形态。也就是说，在明文形态和密文形态的数据混合计算的时候，为了防止密文形态的数据被泄露，应将明文形态的数据也转换为密文形态的数据后再参与计算。

[0073] 基于上述实际应用场景，在一种可行的实施方式中，如图2所示，步骤S2具体包括：

[0074] 步骤S21、获取待处理任务涉及的每个业务数据的初始数据形态；

[0075] 步骤S22、判断是否有任一业务数据的初始数据形态为密文形态，若是，则目标数据形态为密文形态，若否，则目标数据形态为明文形态。

[0076] 在一种可行的实施方式中，步骤S3包括：基于每个业务数据的数据封装对象，调用对应目标数据形态的获取方法获取每个业务数据的目标数据形态。

[0077] 其中，数据封装对象包括业务数据的初始数据形态，以及业务数据不同数据形态的获取方法，以及用于将初始数据形态转换成其他数据形态的预设转换算法。

[0078] 具体地，控制端在接收到原始计算任务时，会创建原始计算任务涉及到的每个业务数据的数据封装对象。并将业务数据的初始数据形态存储在业务数据的数据封装对象中，同时该数据封装对象中还封装了用于获取该业务数据的每一种数据形态的获取方法，以及用于将初始数据形态转换成其他数据形态的预设转换算法。

[0079] 在数据处理节点使用该业务数据时，只需要通过业务数据的数据封装对象调用对应目标数据形态的获取方法，即可直接获得业务数据的目标数据形态。即，调用对应目标数据形态的获取方法后，判断业务数据的目标数据形态是否存在，若存在可以直接获取到业务数据的目标数据形态，若不存在，则调用将初始数据形态转换成目标数据形态的预设转换算法获取业务数据的目标数据形态，并存储在数据封装对象中。

[0080] 在实际应用场景中，如图3所示，以数据形态包括明文形态和密文形态为例，数据封装类定义了两个变量以及四个方法。

[0081] 变量包括明文形态public_data和密文形态private_data。方法包括用于获取明文形态public_data的get_public_data()、用于获取密文形态private_data的get_private_data()、用于设置明文形态的set_public_data(public_data)以及用于设置密文形态的set_private_data(private_data)。

[0082] 控制端在接收到原始计算任务时，会基于上述数据封装类创建每个业务数据的数据封装对象，同时通过set_public_data(public_data)方法将业务数据的初始数据形态赋值给数据封装对象中的变量public_data，或者通过set_private_data(private_data)方法将业务数据的初始数据形态赋值给数据封装对象中的变量private_data。

[0083] 如果数据处理节点需要使用业务数据的明文形态，直接通过数据封装对象中的get_public_data()方法，判断public_data是否存在，若存在，get_public_data()方法直接返回业务数据的明文形态，若不存在，get_public_data()方法内部调用将初始数据形态转换成目标数据形态的预设转换算法获取业务数据的目标数据形态，并通过set_public_data(public_data)方法为变量public_data赋值，进而得到业务数据的明文形态。

[0084] 同理的，如果数据处理节点需要使用业务数据的密文形态，直接通过数据封装对象中的get_private_data()方法，判断private_data是否存在，若存在，get_private_data()方法直接返回业务数据的密文形态，若不存在，get_private_data()方法内部调用将初始数据形态转换成目标数据形态的预设转换算法获取业务数据的目标数据形态，并通过set_private_data(private_data)方法为变量private_data赋值，进而得到业务数据的密文形态。

[0085] 在步骤S4之后，数据处理节点会为待处理任务的计算结果创建对应的数据封装对象，并将计算结果的数据封装对象的标识信息返回给控制端。通过该标识信息从相应的数据处理节点中提取出所需要的数据。

[0086] 实施例2

[0087] 本实施例提供一种数据混合处理系统，数据混合处理系统部署于数据处理节点，数据处理节点部署有多种数据形态对应的计算逻辑。

[0088] 如图4所示，该数据混合处理系统包括：

[0089] 任务接收模块11，用于接收待处理任务，待处理任务涉及至少一个业务数据；

[0090] 数据形态确定模块12，用于根据待处理任务基于预设策略确定目标数据形态；

[0091] 数据获取模块13，用于获取每个业务数据的目标数据形态；

[0092] 任务处理模块14，用于基于目标数据形态的业务数据，采用目标数据形态对应的计算逻辑获取待处理任务的计算结果。

[0093] 任务接收模块11接收的待处理任务，通常为对一个业务数据进行运算或对若干个业务数据进行同级运算。因此，一个待处理任务涉及至少一个业务数据。

[0094] 在一种可行的实施方式中，待处理任务为原始计算任务拆分成的子任务，不同的数据处理节点会接收到不同的子任务，接收不同子任务的数据处理节点串行或并行进行处理。

[0095] 例如，如果多个数据处理节点接收到的待处理任务所涉及的业务数据可以直接获取得到，那么这些数据处理节点可以并行处理对应的待处理任务。如果某些待处理任务所涉及的业务数据需要等待其他数据处理节点返回的计算结果，这些待处理任务会需要的计算结果返回后才被分发至数据处理节点，因此，接收此类待处理任务的数据处理节点相对于为其返回计算结果的数据处理节点来说，是串行进行处理的。

[0096] 数据形态确定模块12用于根据预设策略确定数据处理节点需要在数据处理的过程中使用的目标数据形态。

[0097] 在一种可行的实施方式中，数据形态确定模块12具体用于获取待处理任务涉及的每个业务数据的初始数据形态，并根据每个业务数据的初始数据形态对应的安全级别确定目标数据形态。

[0098] 在另一种可行的实施方式中，数据形态确定模块12具体用于当待处理任务涉及多个业务数据时，判断业务数据是否来自于多个用户，若是，按照用户之间的约定数据形态确定目标数据形态。

[0099] 也就是说，预设策略主要围绕着数据安全进行设置。一种策略是，在待处理任务涉及对应不同安全级别的多个业务数据时，通常选用这些业务数据中最高安全级别对应的数据形态作为目标数据形态。另一种策略是，在待处理任务涉及多个业务数据的基础上，进一

步判断业务数据是否来自不同的用户，如果是，为了防止业务数据在传输或者处理的过程中泄露，将用户之间约定好的数据形态作为目标数据形态。此时的目标数据形态通常是需要加密的。

[0100] 在实际应用场景中，待处理任务涉及第一业务数据和第二业务数据，第一业务数据的初始数据形态为明文形态，第二业务数据的初始数据形态为密文形态。显然地，密文形态的安全级别高于明文形态。因此，目标数据形态应为密文形态。也就是说，在明文形态和密文形态的数据混合计算的时候，为了防止密文形态的数据被泄露，应将明文形态的数据也转换为密文形态的数据后再参与计算。

[0101] 基于上述实际应用场景，在一种可行的实施方式中，如图2所示，数据形态确定模块12具体用于获取待处理任务涉及的每个业务数据的初始数据形态；判断是否有任一业务数据的初始数据形态为密文形态，若是，则目标数据形态为密文形态，若否，则目标数据形态为明文形态。

[0102] 在一种可行的实施方式中，数据获取模块13具体用于基于每个业务数据的数据封装对象，调用对应目标数据形态的获取方法获取每个业务数据的目标数据形态。

[0103] 其中，数据封装对象包括业务数据的初始数据形态，以及业务数据不同数据形态的获取方法，以及用于将初始数据形态转换成其他数据形态的预设转换算法。

[0104] 具体地，控制端在接收到原始计算任务时，会创建原始计算任务涉及到的每个业务数据的数据封装对象。并将业务数据的初始数据形态存储在业务数据的数据封装对象中，同时该数据封装对象中还封装了用于获取该业务数据的每一种数据形态的获取方法，以及用于将初始数据形态转换成其他数据形态的预设转换算法。

[0105] 在数据处理节点使用该业务数据时，只需要通过业务数据的数据封装对象调用对应目标数据形态的获取方法，即可直接获得业务数据的目标数据形态。即，调用对应目标数据形态的获取方法后，判断业务数据的目标数据形态是否存在，若存在可以直接获取到业务数据的目标数据形态，若不存在，则调用将初始数据形态转换成目标数据形态的预设转换算法获取业务数据的目标数据形态，并存储在数据封装对象中。

[0106] 在实际应用场景中，如图3所示，以数据形态包括明文形态和密文形态为例，数据封装类定义了两个变量以及四个方法。

[0107] 变量包括明文形态public_data和密文形态private_data。方法包括用于获取明文形态public_data的get_public_data()、用于获取密文形态private_data的get_private_data()、用于设置明文形态的set_public_data(public_data)以及用于设置密文形态的set_private_data(private_data)。

[0108] 控制端在接收到原始计算任务时，会基于上述数据封装类创建每个业务数据的数据封装对象，同时通过set_public_data(public_data)方法将业务数据的初始数据形态赋值给数据封装对象中的变量public_data，或者通过set_private_data(private_data)方法将业务数据的初始数据形态赋值给数据封装对象中的变量private_data。

[0109] 如果数据处理节点需要使用业务数据的明文形态，直接过数据封装对象中的get_public_data()方法，判断public_data是否存在，若存在，get_public_data()方法直接返回业务数据的明文形态，若不存在，get_public_data()方法内部调用将初始数据形态转换成目标数据形态的预设转换算法获取业务数据的目标数据形态，并通过set_public_

data(public_data)方法为变量public_data赋值,进而得到业务数据的明文形态。

[0110] 同理的,如果数据处理节点需要使用业务数据的密文形态,直接过数据封装对象中的get_private_data()方法,判断private_data是否存在,若存在,get_private_data()方法直接返回业务数据的密文形态,若不存在,get_private_data()方法内部调用将初始数据形态转换成目标数据形态的预设转换算法获取业务数据的目标数据形态,并通过set_private_data(private_data)方法为变量private_data赋值,进而得到业务数据的密文形态。

[0111] 任务处理模块14按照对应目标数据形态的计算逻辑获取到待处理任务的计算结果后,会为该计算结果创建相应的数据封装对象,并将计算结果的数据封装对象的标识信息返回给控制端。通过该标识信息从相应的数据处理节点中提取出所需要的数据。

[0112] 实施例3

[0113] 本实施例提供一种数据处理系统。

[0114] 如图5所示,数据处理系统包括控制端20和若干数据处理节点21。

[0115] 其中,控制端20用于为数据处理节点21发送待处理任务。

[0116] 具体地,控制端20接收原始计算任务,将原始计算任务拆分成多个子任务,并将每个子任务作为待处理任务分发给数据处理节点21。

[0117] 其中,数据处理节点21部署有多种数据形态对应的计算逻辑以及实施例2及其任一种实施方式的数据混合处理系统。

[0118] 下面以用于明密文数据混合处理的数据处理系统为例。

[0119] 传统的用于明密文数据混合处理的数据处理系统,其架构如图6所示,包括控制端、1至N个明文节点以及1至M个密文节点。明文节点只能用于处理明文形态的数据。明文计算比较简单,所以计算效率较高,适用于用户本地进行一些数据计算,但不适用于对数据安全级别较高的计算场景。密文节点只能用于处理密文形态的数据。由于数据想要以密文形态参与的计算过程,需要进行加密,并且基于密文形态的计算会提高复杂度,所以计算效率较低,适用于对数据安全级别较高的计算场景。

[0120] 而基于本实施例提供的数据处理系统,其架构如图7所示,节点上同时部署有用于处理明文形态和密文形态的数据计算逻辑,因此,不再需要区分明文节点和密文节点。一方面可以减少系统的部署过程和部署难度,另一方面还可以避免过多节点闲置,提供节点利用率。

[0121] 控制端收到某一次计算任务时,会将涉及到的业务数据都封装成数据DS对象,同时将该计算任务拆分成n个子任务,并行或串行地下发到各个节点。节点收到某个子任务后,根据预设策略决定是进行明文计算还是密文计算,随后调用对应的明文或者密文的计算逻辑进行计算。

[0122] 其中,如果某个子任务采用的是明文计算,那么节点中部署的数据混合处理系统通过数据DS对象的get_public_data()方法来获取业务数据的明文形态;如果采用的是密文计算,那节点中部署的数据混合处理系统通过数据DS对象的get_private_data()方法来获取业务数据的密文形态。

[0123] 然后根据相应的计算逻辑,计算该子任务。计算完成后,为计算结果也创建一个数据DS对象,如果计算结果的数据是明文形态,就通过数据DS对象调用set_public_data()

方法来设置该计算结果的明文形态，如果结算结果的数据是密文形态，就通过数据DS对象调用set_private_data()方法来设置该计算结果的密文形态，返回给控制端该计算结果对应的数据DS对象的标识信息。

[0124] 如果此次子任务还有后续子任务，控制端再次将后续的子任务分发到各个节点继续计算，直至整个完整的计算任务完成。整个计算任务完成后，控制端调用最后返回的数据DS对象的get_public_data()方法就可以获得本次计算任务的最终明文结果数据。

[0125] 示例性的，在实际应用场景中，计算任务为(D1+3)×(D2+5)，D1和D2分别表示公司A和公司B的业务数据。如图8所示，公司A和公司B中各部署有数据处理系统，公司A部署有第一控制端201以及1至N个第一数据处理节点211，公司B部署有第二控制端202以及1至M个第二数据处理节点212。

[0126] 如果计算任务的发起方为公司A，那么公司A的第一控制端201将计算任务分解成三个子任务。第一个子任务为D1+3，结果定义为DA；第二个子任务为D2+5，结果定义为DB；第三个子任务为DA×DB，结果定义为DC。因为D1+3和D2+5在计算时不具有前后依赖关系，因此可以同步进行。所以同时将D1+3和D2+5分发给不同的节点进行计算。

[0127] 如果公司A和公司B部署同一套系统，那么第一控制端201可以直接将第一个子任务分发给第一数据处理节点211以及将第二个子任务分发给第二数据处理节点212。如果公司A和公司B部署不同的系统，第一控制端201将第一个子任务分发第一数据处理节点211，并向第二控制端202发送第二个子任务。第二控制端202在接收到第二个子任务后，分发给第二数据处理节点212进行计算。

[0128] 以公司A和公司B部署同一套系统为例：

[0129] 在公司A中的数据处理节点1进行计算时，如果D1的初始数据形态为明文形态，D1+3是明文形态的数据与一个常数相加，因此属于部署在公司A中的数据处理节点内部操作，即使用明文形态的数据计算逻辑也不会造成数据泄露，因此，目标数据形态确定为明文形态，计算结果DA也为明文形态，并为计算结果DA创建相应的数据DS对象，通过数据DS对象调用set_public_data(public_data)方法将计算结果DA的明文形态进行保存，并将DA的数据DS对象的标识信息返回给第一控制端201。

[0130] 在公司B中的数据处理节点1进行计算时，如果D2的初始数据形态也为明文形态，同理的，也属于部署在公司B中的数据处理节点内部操作，即可采用明文形态的数据计算逻辑计算D2+5，得到计算结果DB，并将计算结果DB的明文形态保存至对应的数据DS对象，并将DB的数据DS对象的标识信息返回给第一控制端201。

[0131] 即，DA的数据DS对象是公司A中的数据处理节点1的临时数据，通常保存在内存中。DB的数据DS对象同理。通过标识信息可以数据处理节点中获取相应的数据DS对象进行使用。

[0132] 第一控制端201在接收到DA的数据DS对象的标识信息以及DB的数据DS对象的标识信息后，就可以分发第三个子任务，第三个子任务携带了DA的数据DS对象的标识信息以及DB的数据DS对象的标识信息。

[0133] 一种实施方式是将DA×DB的子任务调度到计算任务再次下发给公司A中的数据处理节点2。

[0134] 公司A中的数据处理节点2在计算第三个子任务时，由于DB在公司B的系统中，因此

需要采用密文形态进行计算。具体地，公司A中的数据处理节点2会通过DA的数据DS对象的标识信息从公司A中的数据处理节点1获取到DA的数据DS对象，并通过DA的数据DS对象中的get_private_data()方法获取DA的密文形态。同理的，公司A中的数据处理节点2会通过DB的数据DS对象的标识信息通知公司B中的数据处理节点1，由公司B中的数据处理节点1通过DB的数据DS对象中的get_private_data()方法获取DB的密文形态并返回给公司A中的数据处理节点2。然后基于DA的密文形态以及DB的密文形态按照密文形态的数据计算逻辑获取DA×DB的计算结果DC。此时的DC也是密文形态，公司A中的数据处理节点2会为DC创建相应的数据DS对象，并通过DC的数据DS对象调用set_private_data(private_data)将DC的密文形态保存至DC的数据DS对象中，最后将DC的数据DS对象的标识信息返回给第一控制端201。

[0135] 第一控制端201便可以根据DC的数据DS对象的标识信息，通知公司A中的数据处理节点2，由公司A中的数据处理节点2在DC的数据DS对象中的get_public_data()方法即可获取到DC的明文形态，随后将计算结果DC的明文形态返回给201控制端，最后返回给公司A的用户。

[0136] 另一种实施方式是采用秘密分享技术。例如，将DA分成DA1和DA2，使得DA＝DA1+DA2，DB同理，分成DB1和DB2。然后公司A中的数据处理节点1获得DB1，公司B中的数据处理节点1获得DA2，随后通过某种约定的秘密分享协议，在公司A中的数据处理节点1计算得到DC1，在公司B中的数据处理节点1计算得到DC2，最后公司A中的数据处理节点2根据DC1和DC2计算得到DC。这样公司A无法获取完整的DB，可以保障DB的安全，同理的，公司B也无法获取完成DA，也保证了DA的安全。

[0137] 本实施例提供的数据处理系统中的数据处理节点21，由于部署了多种数据形态对应的计算逻辑，每个数据处理节点21都可以进行不同数据形态的业务数据计算。不再需要将不同数据形态的计算逻辑分开管理和审计，有利于数据混合处理系统的开发和管理。并且，在数据处理节点21上部署可以处理不同数据形态的业务数据的数据混合处理系统，可以极大程度地提高数据处理系统的并发能力，提高原始计算任务的子任务的处理速度，进而提升原始计算任务的成功率。另外，在部署数据处理系统时，不再需要部署用于处理不同数据形态的数据处理节点21，简化系统部署过程。

[0138] 特别是，一个待处理任务中涉及到多种数据形态的数据混合处理时，控制端20不再需要明确区分待处理任务使用数据的数据形态，也不再需要显示地调用数据形态之间的转换方法，极大程度地减少了控制端20的开发难度。

[0139] 实施例4

[0140] 图9为本公开实施例4提供的一种电子设备的结构示意图。电子设备包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，处理器执行程序时，实现实施例1及其任一种实施方式所述的数据混合处理方法。图9显示的电子设备30仅仅是一个示例，不应对本公开实施例的功能和使用范围带来任何限制。

[0141] 如图9所示，电子设备30可以通用计算设备的形式表现，例如其可以为服务器设备。电子设备30的组件可以包括但不限于：上述至少一个处理器31、上述至少一个存储器32、连接不同系统组件(包括存储器32和处理器31)的总线33。

[0142] 总线33包括数据总线、地址总线和控制总线。

[0143] 存储器32可以包括易失性存储器，例如随机存取存储器(RAM)321和/或高速缓存存储器322，还可以进一步包括只读存储器(ROM)323。

[0144] 存储器32还可以包括具有一组(至少一个)程序模块324的程序/实用工具325，这样的程序模块324包括但不限于：操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0145] 处理器31通过运行存储在存储器32中的计算机程序，从而执行各种功能应用以及数据处理，例如本公开实施例1及其任一种实施方式所述的数据混合处理方法。

[0146] 电子设备30也可以与一个或多个外部设备34(例如键盘、指向设备等)通信。这种通信可以通过输入/输出(I/O)接口35进行。并且，模型生成的设备30还可以通过网络适配器36与一个或者多个网络(例如局域网(LAN)，广域网(WAN)和/或公共网络，例如因特网)通信。如图9所示，网络适配器36通过总线33与模型生成的设备30的其它模块通信。应当明白，尽管图中未示出，可以结合模型生成的设备30使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理器、外部磁盘驱动阵列、RAID(磁盘阵列)系统、磁带驱动器以及数据备份存储系统等。

[0147] 应当注意，尽管在上文详细描述中提及了电子设备的若干单元/模块或子单元/模块，但是这种划分仅仅是示例性的并非强制性的。实际上，根据本公开的实施方式，上文描述的两个或更多单元/模块的特征和功能可以在一个单元/模块中具体化。反之，上文描述的一个单元/模块的特征和功能可以进一步划分为由多个单元/模块来具体化。

[0148] 实施例5

[0149] 本实施例提供了一种计算机可读存储介质，其上存储有计算机程序，程序被处理器执行时，实现实施例1及其任一种实施方式所述的数据混合处理方法。

[0150] 其中，可读存储介质可以采用的更具体可以包括但不限于：便携式盘、硬盘、随机存取存储器、只读存储器、可擦拭可编程只读存储器、光存储器件、磁存储器件或上述的任意合适的组合。

[0151] 在可能的实施方式中，本公开还可以实现为一种程序产品的形式，其包括程序代码，当程序产品在终端设备上运行时，程序代码用于使终端设备执行时，实现实施例1及其任一种实施方式所述的数据混合处理方法。

[0152] 其中，可以一种或多种程序设计语言的任意组合来编写用于执行本公开的程序代码，程序代码可以完全地在用户设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户设备上部分在远程设备上执行或完全在远程设备上执行。

[0153] 虽然以上描述了本公开的具体实施方式，但是本领域的技术人员应当理解，这仅是举例说明，本公开的保护范围是由所附权利要求书限定的。本领域的技术人员在不背离本公开的原理和实质的前提下，可以对这些实施方式做出多种变更或修改，但这些变更和修改均落入本公开的保护范围。

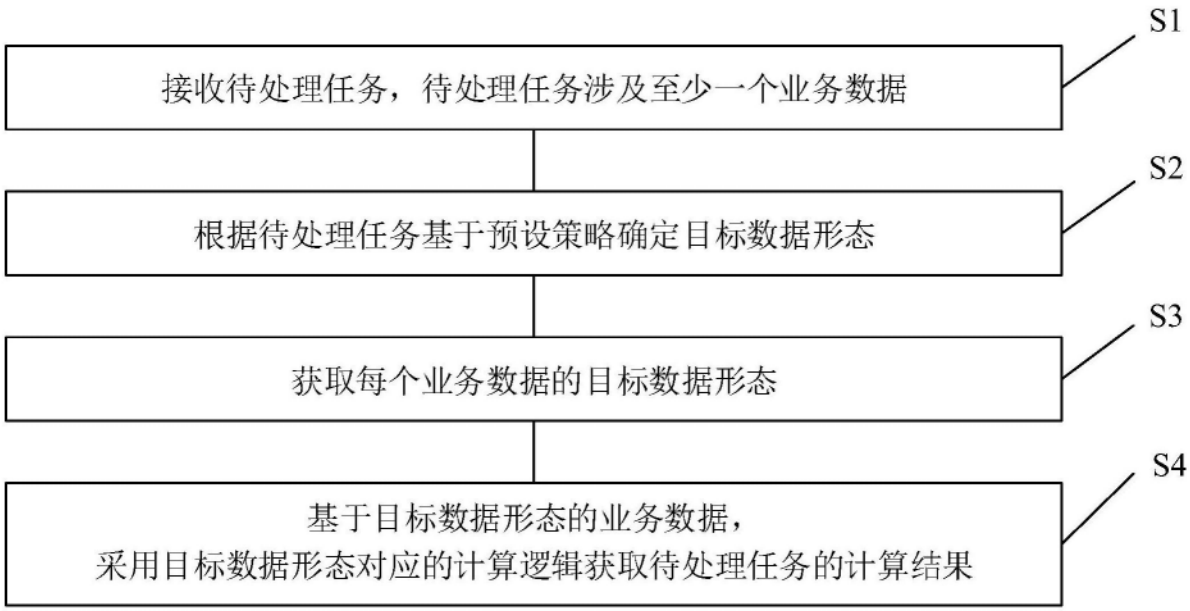

S1
接收待处理任务，待处理任务涉及至少一个业务数据
S2
根据待处理任务基于预设策略确定目标数据形态
S3
获取每个业务数据的目标数据形态
S4
基于目标数据形态的业务数据，
采用目标数据形态对应的计算逻辑获取待处理任务的计算结果

图1

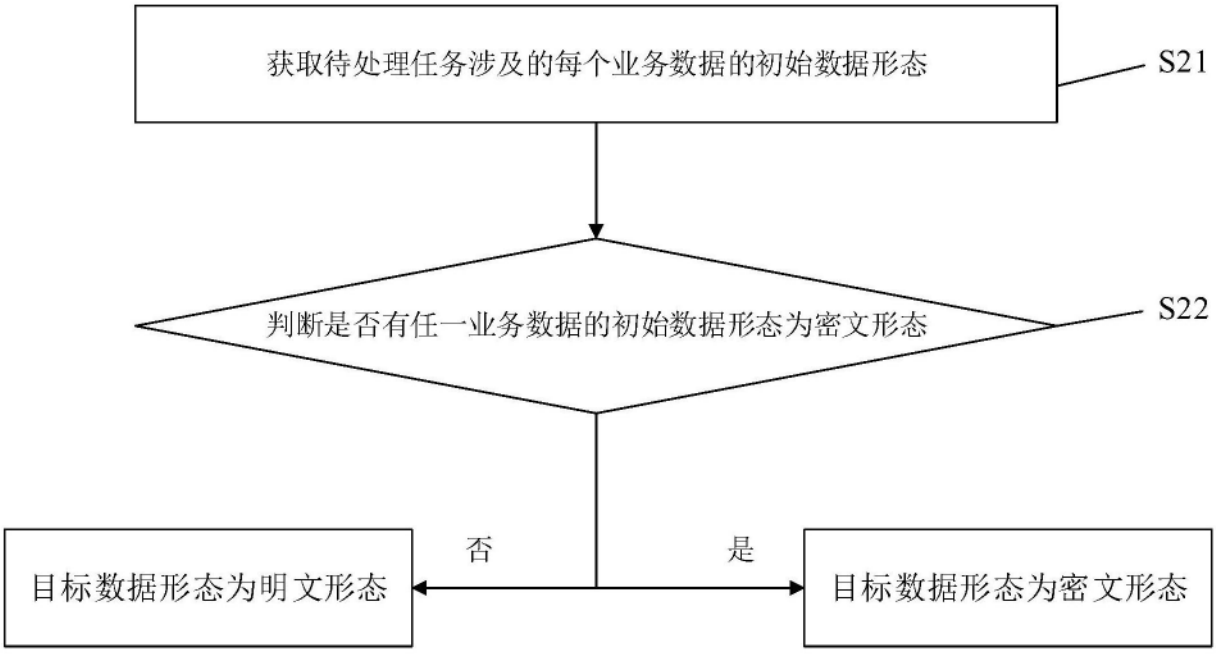

获取待处理任务涉及的每个业务数据的初始数据形态
S21
判断是否有任一业务数据的初始数据形态为密文形态
S22
否
是
目标数据形态为明文形态
目标数据形态为密文形态

图2

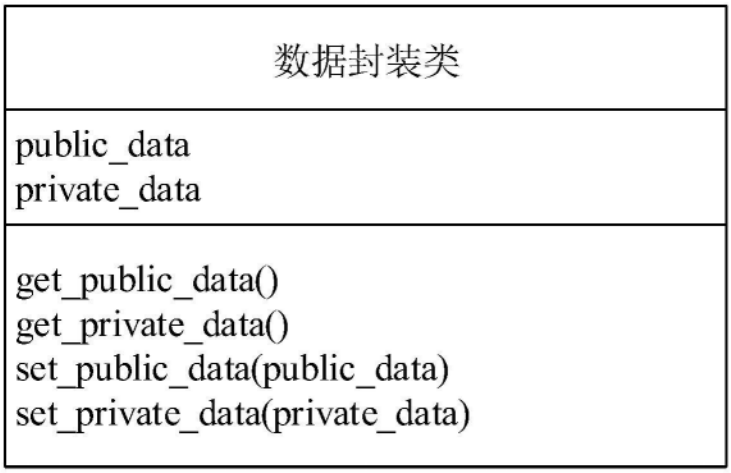

数据封装类
public_data
private_data
get_public_data()
get_private_data()
set_public_data(public_data)
set_private_data(private_data)

图3

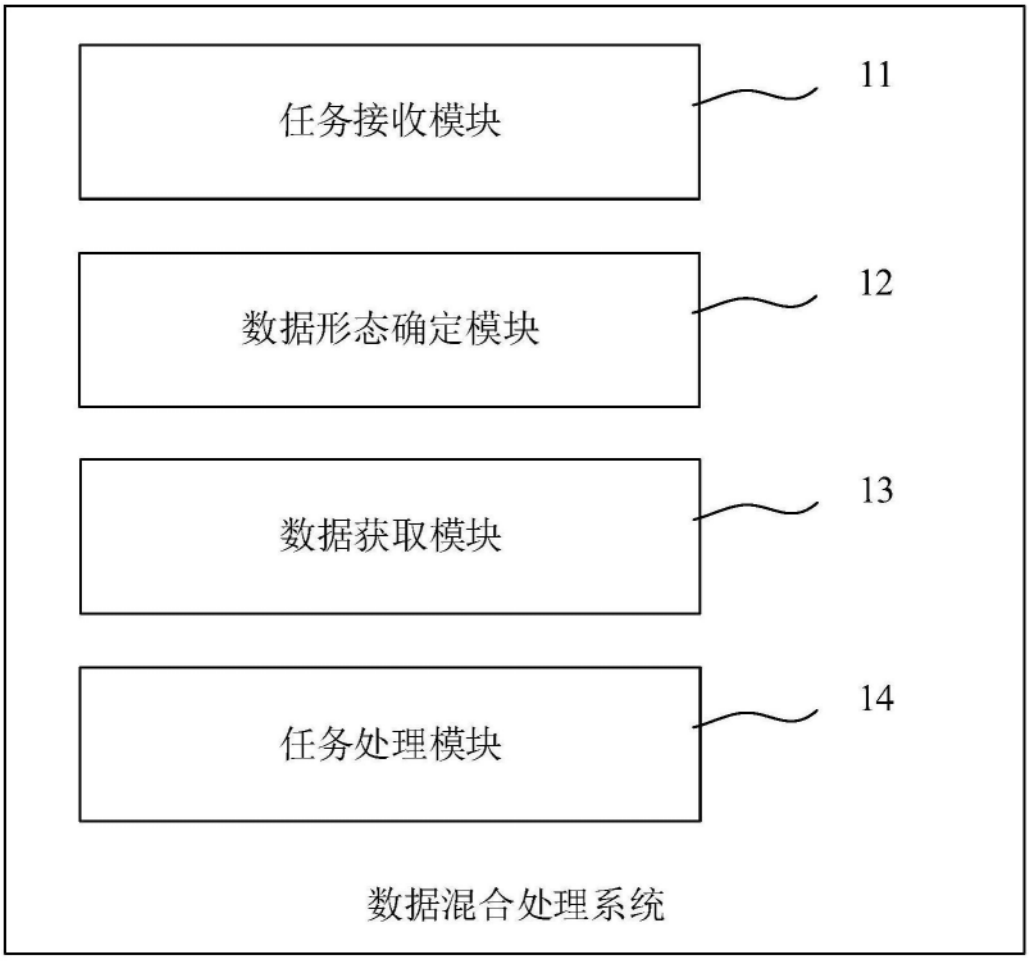

任务接收模块
11
数据形态确定模块
12
数据获取模块
13
任务处理模块
14
数据混合处理系统

图4

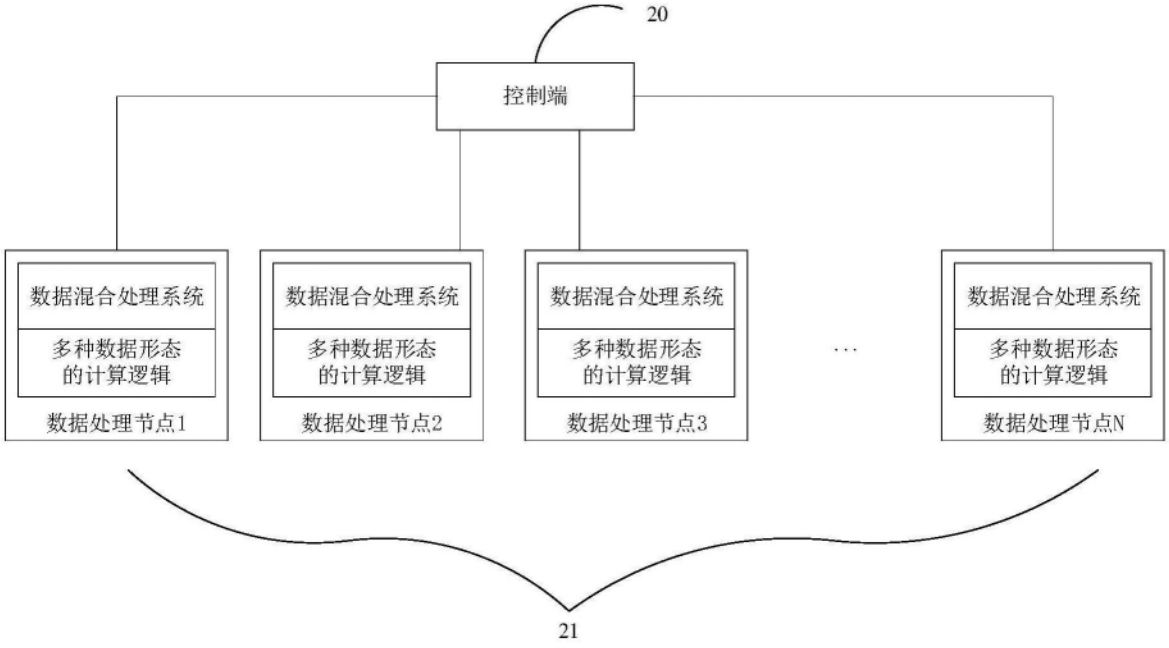

20
控制端
数据混合处理系统
多种数据形态的计算逻辑
数据处理节点1
数据混合处理系统
多种数据形态的计算逻辑
数据处理节点2
数据混合处理系统
多种数据形态的计算逻辑
数据处理节点3
...
数据混合处理系统
多种数据形态的计算逻辑
数据处理节点N
21

图5

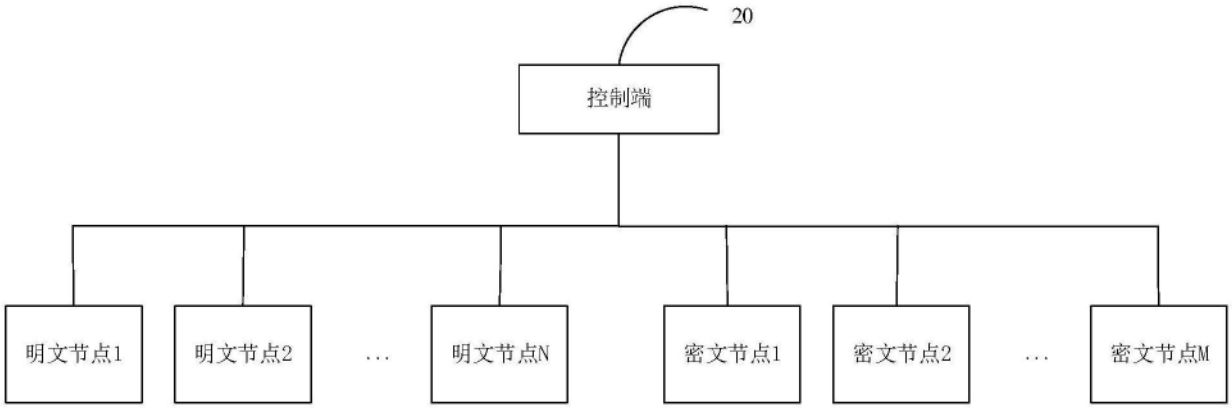

20
控制端
明文节点1
明文节点2
...
明文节点N
密文节点1
密文节点2
...
密文节点M

图6

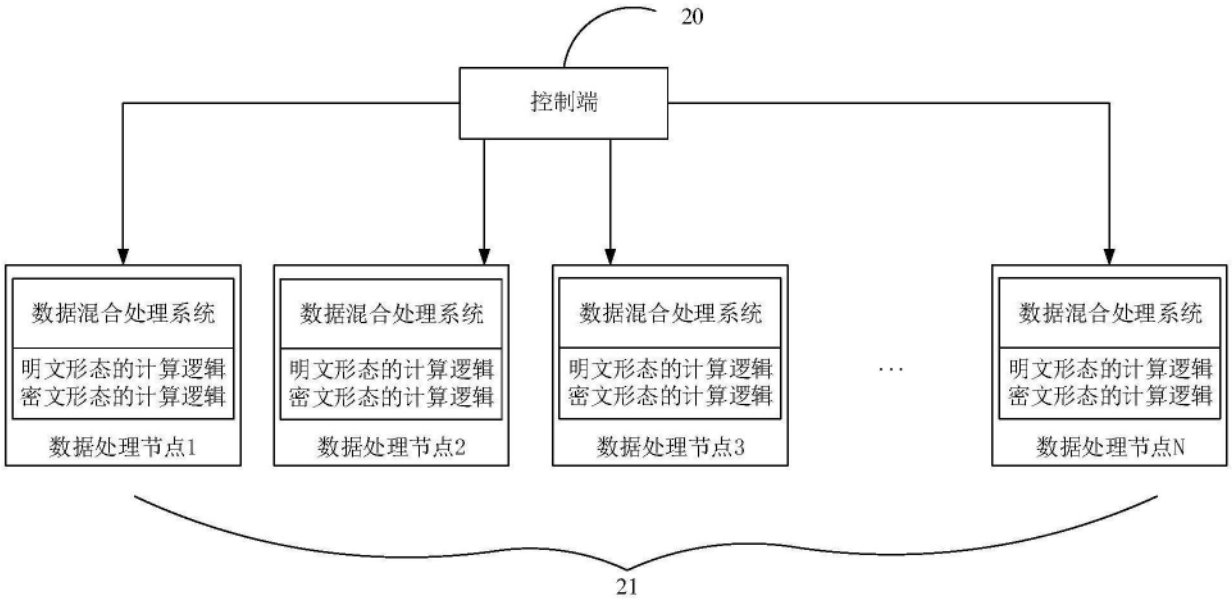

20
控制端
数据混合处理系统
明文形态的计算逻辑
密文形态的计算逻辑
数据处理节点1
数据混合处理系统
明文形态的计算逻辑
密文形态的计算逻辑
数据处理节点2
数据混合处理系统
明文形态的计算逻辑
密文形态的计算逻辑
数据处理节点3
...
数据混合处理系统
明文形态的计算逻辑
密文形态的计算逻辑
数据处理节点N
21

图7

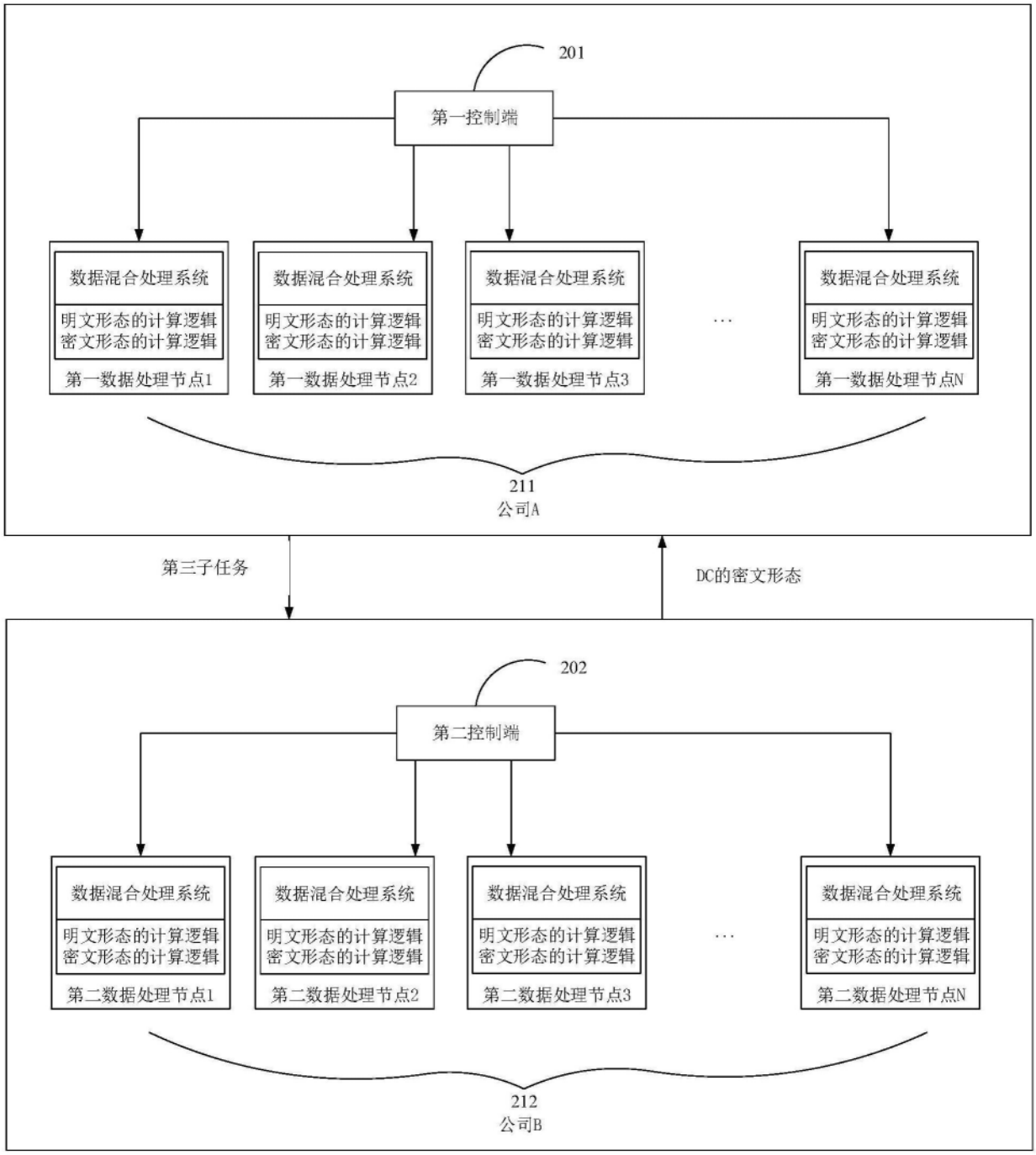

201
第一控制端
数据混合处理系统
明文形态的计算逻辑
密文形态的计算逻辑
第一数据处理节点1
第一数据处理节点2
第一数据处理节点3
…
第一数据处理节点N
211
公司A
第三子任务
DC的密文形态
202
第二控制端
第二数据处理节点1
第二数据处理节点2
第二数据处理节点3
第二数据处理节点N
212
公司B

图8

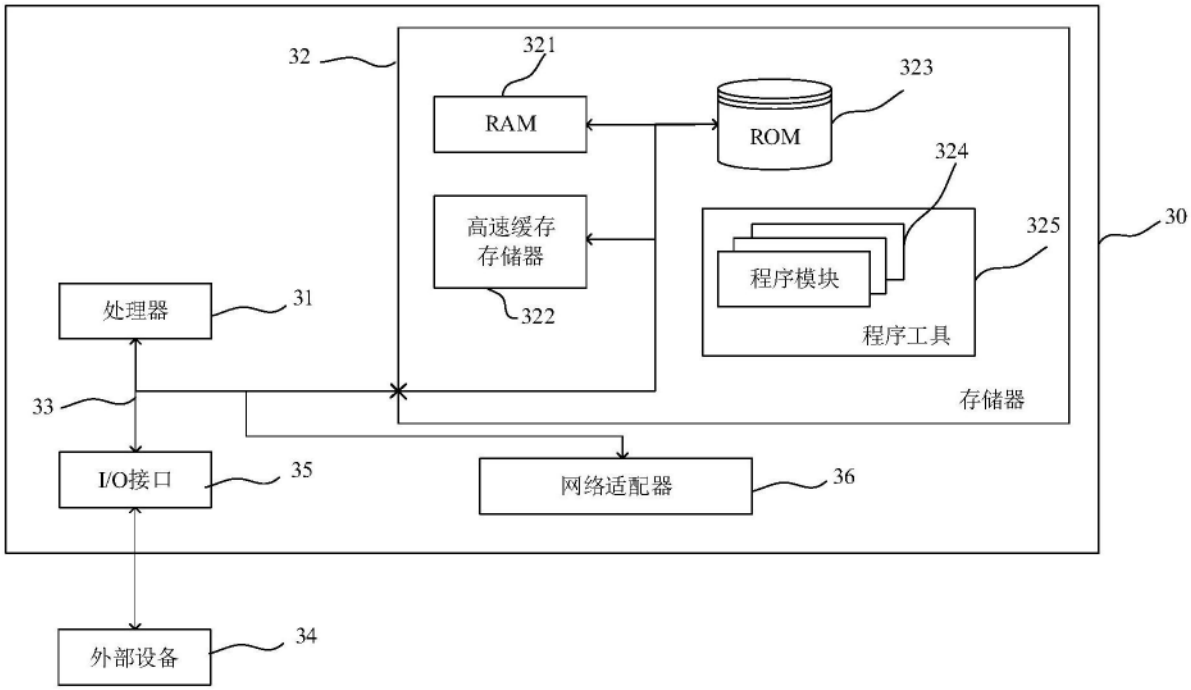

32
321
RAM
323
ROM
324
高速缓存
存储器
322
325
30
程序模块
程序工具
处理器
31
33
存储器
I/O接口
35
网络适配器
36
外部设备
34

图9