(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105335218 A
(43)申请公布日 2016.02.17

(21)申请号 201410317237.2

(22)申请日 2014.07.03

(71)申请人 北京金山安全软件有限公司
地址 100085 北京市海淀区小营西路33号二层东区

(72)发明人 潘洪安 张楠

(74)专利代理机构 广州三环专利代理有限公司 44202
代理人 郝传鑫 熊永强

(51)Int.Cl.
G06F 9/46(2006.01)
G06F 9/50(2006.01)

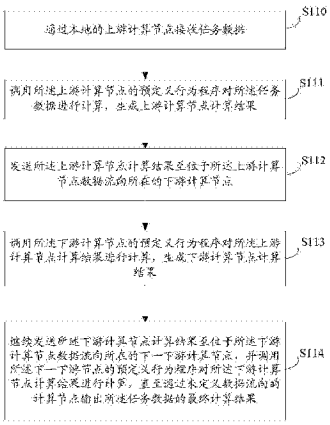
权利要求书3页 说明书9页 附图2页

(54)发明名称

一种基于本地的流式计算方法及流式计算系统

(57)摘要

本发明实施例公开了一种基于本地的流式计算方法及流式计算系统，包括：通过本地的上游计算节点接收任务数据；调用上游计算节点的预定义行为程序对任务数据进行计算，生成上游计算节点计算结果；发送上游计算节点计算结果至位于上游计算节点数据流向所在的下游计算节点；调用下游计算节点的预定义行为程序对上游计算节点计算结果进行计算，生成下游计算节点计算结果；继续发送下游计算节点计算结果至位于下游计算节点数据流向所在的下一下游计算节点，并调用下一下游节点的预定义行为程序对下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出任务数据的最终计算结果。实施本发明，能够提高计算效率。



CN 105335218 A

1. 一种基于本地的流式计算方法，其特征在于，包括：

通过本地的上游计算节点接收任务数据；

调用所述上游计算节点的预定义行为程序对所述任务数据进行计算，生成上游计算节点计算结果；

发送所述上游计算节点计算结果至位于所述上游计算节点数据流向所在的下游计算节点；

调用所述下游计算节点的预定义行为程序对所述上游计算节点计算结果进行计算，生成下游计算节点计算结果；

继续发送所述下游计算节点计算结果至位于所述下游计算节点数据流向所在的下一下游计算节点，并调用所述下一下游节点的预定义行为程序对所述下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出所述任务数据的最终计算结果。

2. 如权利要求 1 所述的方法，其特征在于，通过本地的上游计算节点接收任务数据之前，还包括：

为所述本地中的各计算节点设置所述预定义行为程序。

3. 如权利要求 2 所述的方法，其特征在于，为所述本地中的节点设置预定义行为程序之前，通过本地的上游计算节点接收任务数据之后，还包括：

定义所述各计算节点之间的数据流向关系。

4. 如权利要求 1 所述的方法，其特征在于，所述上游计算节点接收到的为多个任务数据，调用所述上游计算节点的预定义行为程序对所述任务数据进行计算，生成上游计算节点计算结果包括：

将多个所述任务数据插入节点输入队列；

按照所述节点输入队列中所述任务数据的排列顺序分配多个所述任务数据至所述上游计算节点的计算单元；

使所述计算单元计算所述任务数据，所述上游计算节点包含至少一个所述计算单元；

将所述至少一个计算单元计算出的计算结果插入节点输出队列，形成所述上游计算节点计算结果。

5. 如权利要求 4 所述的方法，其特征在于，所述计算单元分配到的为多个任务数据，所述计算单元为多线程式计算单元，使所述计算单元计算所述任务数据包括：

为分配到所述计算单元的多个任务数据分配计算线程；

使所述计算线程对分配到的所述任务数据进行计算。

6. 如权利要求 5 所述的方法，其特征在于，所述计算线程分配到的为多个任务数据，使所述计算线程对分配到的所述任务数据进行计算包括：

将分配到所述计算线程的多个所述任务数据插入线程输入队列；

按照所述线程输入队列中所述任务数据的排列顺序对所述任务数据进行计算，生成所述计算结果。

7. 如权利要求 4 所述的方法，其特征在于，所述上游计算节点包含指定数目的所述计算单元。

8. 如权利要求 5 所述的方法，其特征在于，所述计算单元包含指定数目的所述计算线程。

9. 如权利要求 2 所述的方法，其特征在于，所述各计算节点的所述预定义行为程序是从应用程序编程接口 API 输入的。

10. 如权利要求 3 所述的方法，其特征在于，所述各计算节点之间的数据流向关系是根据配置文件定义的。

11. 一种流式计算系统，其特征在于，包括：

上游计算节点，位于本地，用于接收任务数据；

所述上游计算节点，还用于调用所述上游计算节点的预定义行为程序对所述任务数据进行计算，生成上游计算节点计算结果；

所述上游计算节点，还用于发送所述上游计算节点计算结果至位于所述上游计算节点数据流向所在的下游计算节点；

下游计算节点，为所述上游计算节点的数据流向所在，用于接收所述上游计算节点发送的所述上游计算节点计算结果；

所述下游计算节点，还用于调用所述下游计算节点的预定义行为程序对所述上游计算节点计算结果进行计算，生成下游计算节点计算结果；

所述下游计算节点，还用于继续发送所述下游计算节点计算结果至位于所述下游计算节点数据流向所在的下一下游计算节点，使所述下一下游节点调用所述下一下游节点的预定义行为程序对所述下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出所述任务数据的最终计算结果。

12. 如权利要求 11 所述的流式计算系统，其特征在于，还包括：

计算节点管理器，用于为所述本地中的各计算节点设置所述预定义行为程序。

13. 如权利要求 12 所述的流式计算系统，其特征在于，

所述计算节点管理器，还用于定义所述各计算节点之间的数据流向关系。

14. 如权利要求 11 所述的流式计算系统，其特征在于，所述上游计算节点接收到的为多个任务数据，

所述上游计算节点，还用于将多个所述任务数据插入节点输入队列；

所述上游计算节点，还用于按照所述节点输入队列中所述任务数据的排列顺序分配多个所述任务数据至所述上游计算节点的计算单元；

所述上游计算节点包括：

计算单元，用于计算所述任务数据，所述上游计算节点包含至少一个所述计算单元；

所述计算单元，还用于将所述至少一个计算单元计算出的计算结果插入节点输出队列，形成所述上游计算节点计算结果。

15. 如权利要求 14 所述的流式计算系统，其特征在于，所述计算单元分配到的为多个任务数据，所述计算单元为多线程式计算单元，

所述计算单元，还用于为分配到所述计算单元的多个任务数据分配计算线程；

所述计算单元包括：

计算线程，用于计算所述计算单元分配到的所述任务数据。

16. 如权利要求 15 所述的流式计算系统，其特征在于，所述计算线程分配到的为多个任务数据，

所述计算线程，还用于将分配到所述计算线程的多个所述任务数据插入线程输入队

列；还用于按照所述线程输入队列中所述任务数据的排列顺序对所述任务数据进行计算，生成所述计算结果。

17. 如权利要求 14 所述的流式计算系统，其特征在于，所述上游计算节点包含指定数目的所述计算单元。

18. 如权利要求 15 所述的流式计算系统，其特征在于，所述计算单元包含指定数目的所述计算线程。

19. 如权利要求 12 所述的流式计算系统，其特征在于，所述各计算节点的所述预定义行为程序是从应用程序编程接口 API 输入的。

20. 如权利要求 13 所述的流式计算系统，其特征在于，所述各计算节点之间的数据流向关系是根据配置文件定义的。

一种基于本地的流式计算方法及流式计算系统

技术领域

[0001] 本发明涉及计算机技术领域，尤其涉及一种基于本地的流式计算方法及流式计算系统。

背景技术

[0002] 本地在处理多个任务数据时，通常为单机的多线程处理，每个线程同一时间均只能运行一个任务数据，并且直到当前的任务数据运行完毕才能处理下一个任务数据，故，当任务数据的计算量较大时，会长期占用该线程，以致该线程的输入队列中的其他任务数据不能被及时处理；当任务数据包含多项行为的计算时，维护人员需要为该任务数据编写包含多项行为的预定义行为程序，然而，编写的预定义行为程序中的各项行为只能适用于当前的任务数据，进行其他任务数据时，维护人员需要再编写另一个预定义行为程序，非常的耗费精力，也造成了预定义行为程序的资源浪费。

发明内容

[0003] 本发明实施例提供一种基于本地的流式计算方法及流式计算系统，可将处理任务数据的多项行为分布到本地的各个计算节点，通过使各计算节点调用各节点的预定义行为程序并行、协同地处理任务数据，能够提高计算效率，避免程序资源浪费。

[0004] 本发明实施例提供了一种基于本地的流式计算方法，其可包括：

[0005] 通过本地的上游计算节点接收任务数据；

[0006] 调用所述上游计算节点的预定义行为程序对所述任务数据进行计算，生成上游计算节点计算结果；

[0007] 发送所述上游计算节点计算结果至位于所述上游计算节点数据流向所在的下游计算节点；

[0008] 调用所述下游计算节点的预定义行为程序对所述上游计算节点计算结果进行计算，生成下游计算节点计算结果；

[0009] 继续发送所述下游计算节点计算结果至位于所述下游计算节点数据流向所在的下一下游计算节点，并调用所述下一下游节点的预定义行为程序对所述下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出所述任务数据的最终计算结果。

[0010] 其中，所述方法，还包括：

[0011] 为所述本地中的各计算节点设置所述预定义行为程序。

[0012] 其中，为所述本地中的节点设置预定义行为程序之前，通过本地的上游计算节点接收任务数据之后，还包括：

[0013] 定义所述各计算节点之间的数据流向关系。

[0014] 其中，所述上游计算节点接收到的为多个任务数据，调用所述上游计算节点的预定义行为程序对所述任务数据进行计算，生成上游计算节点计算结果包括：

[0015] 将多个所述任务数据插入节点输入队列；

[0016] 按照所述节点输入队列中所述任务数据的排列顺序分配多个所述任务数据至所述上游计算节点的计算单元；

[0017] 使所述计算单元计算所述任务数据，所述上游计算节点包含至少一个所述计算单元；

[0018] 将所述至少一个计算单元计算出的计算结果插入节点输出队列，形成所述上游计算节点计算结果。

[0019] 其中，所述计算单元分配到的为多个任务数据，所述计算单元为多线程式计算单元，使所述计算单元计算所述任务数据包括：

[0020] 为分配到所述计算单元的多个任务数据分配计算线程；

[0021] 使所述计算线程对分配到的所述任务数据进行计算。

[0022] 其中，所述计算线程分配到的为多个任务数据，使所述计算线程对分配到的所述任务数据进行计算包括：

[0023] 将分配到所述计算线程的多个所述任务数据插入线程输入队列；

[0024] 按照所述线程输入队列中所述任务数据的排列顺序对所述任务数据进行计算，生成所述计算结果。

[0025] 其中，所述上游计算节点包含指定数目的所述计算单元。

[0026] 其中，所述计算单元包含指定数目的所述计算线程。

[0027] 其中，所述各计算节点的所述预定义行为程序是从应用程序编程接口 API 输入的。

[0028] 其中，所述各计算节点之间的数据流向关系是根据配置文件定义的。

[0029] 本发明实施例还提供了一种流式计算系统，其可包括：

[0030] 上游计算节点，位于本地，用于接收任务数据；

[0031] 所述上游计算节点，还用于调用所述上游计算节点的预定义行为程序对所述任务数据进行计算，生成上游计算节点计算结果；

[0032] 所述上游计算节点，还用于发送所述上游计算节点计算结果至位于所述上游计算节点数据流向所在的下游计算节点；

[0033] 下游计算节点，为所述上游计算节点的数据流向所在，用于接收所述上游计算节点发送的所述上游计算节点计算结果；

[0034] 所述下游计算节点，还用于调用所述下游计算节点的预定义行为程序对所述上游计算节点计算结果进行计算，生成下游计算节点计算结果；

[0035] 所述下游计算节点，还用于继续发送所述下游计算节点计算结果至位于所述下游计算节点数据流向所在的下一下游计算节点，使所述下一下游节点调用所述下一下游节点的预定义行为程序对所述下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出所述任务数据的最终计算结果。

[0036] 其中，所述流式计算系统，还包括：

[0037] 计算节点管理器，用于为所述本地中的各计算节点设置所述预定义行为程序。

[0038] 其中，所述计算节点管理器，还用于定义所述各计算节点之间的数据流向关系。

[0039] 其中，所述上游计算节点接收到的为多个任务数据，

[0040] 所述上游计算节点，还用于将多个所述任务数据插入节点输入队列；

[0041] 所述上游计算节点，还用于按照所述节点输入队列中所述任务数据的排列顺序分配多个所述任务数据至所述上游计算节点的计算单元；

[0042] 所述上游计算节点包括：

[0043] 计算单元，用于计算所述任务数据，所述上游计算节点包含至少一个所述计算单元；

[0044] 所述计算单元，还用于将所述至少一个计算单元计算出的计算结果插入节点输出队列，形成所述上游计算节点计算结果。

[0045] 其中，所述计算单元分配到的为多个任务数据，所述计算单元为多线程式计算单元，

[0046] 所述计算单元，还用于为分配到所述计算单元的多个任务数据分配计算线程；

[0047] 所述计算单元包括：

[0048] 计算线程，用于计算所述计算单元分配到的所述任务数据。

[0049] 其中，所述计算线程分配到的为多个任务数据，

[0050] 所述计算线程，还用于将分配到所述计算线程的多个所述任务数据插入线程输入队列；还用于按照所述线程输入队列中所述任务数据的排列顺序对所述任务数据进行计算，生成所述计算结果。

[0051] 其中，所述上游计算节点包含指定数目的所述计算单元。

[0052] 其中，所述各计算单元包含指定数目的所述计算线程。

[0053] 其中，所述各计算节点的所述预定义行为程序是从应用程序编程接口 API 输入的。

[0054] 其中，所述选定的计算节点之间的数据流向关系是根据配置文件定义的。

[0055] 在本发明实施例中，系统可将处理任务数据的多项行为分布到本地的各个计算节点，通过使各计算节点调用各节点的预定义行为程序并行、协同地处理任务数据，能够提高计算效率，避免程序资源浪费，实现轻量化计算。

附图说明

[0056] 为了更清楚地说明本发明实施例中的技术方案，下面将对实施例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一些实施例，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，还可以根据这些附图获得其他的附图。

[0057] 图 1 是本发明实施例提供的基于本地的流式计算方法的实施例流程示意图；

[0058] 图 2 是本发明实施例提供的流式计算系统的实施例结构示意图。

具体实施方式

[0059] 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作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。

[0060] 流式计算通常应用于集群系统，其具有分步、有向无环的计算特点，可以大量地处

理集群服务器系统中的任务。为了解决背景技术问题，本发明实施例将流式计算的特点引入了本地的任务数据处理中，以提高任务数据的处理效率。

[0061] 参见图 1，是本发明实施例提供的基于本地的流式计算方法的实施例流程示意图。本实施例中所描述的流式计算方法，包括步骤：

[0062] 步骤 S110，通过本地的上游计算节点接收任务数据。

[0063] 步骤 S111，调用上游计算节点的预定义行为程序对任务数据进行计算，生成上游计算节点计算结果；

[0064] 步骤 S112，发送上游计算节点计算结果至位于上游计算节点数据流向所在的下游计算节点；

[0065] 步骤 S113，调用下游计算节点的预定义行为程序对上游计算节点计算结果进行计算，生成下游计算节点计算结果；

[0066] 步骤 S114，继续发送下游计算节点计算结果至位于下游计算节点数据流向所在的下一下游计算节点，并调用下一下游节点的预定义行为程序对下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出任务数据的最终计算结果。

[0067] 具体实现中，本发明实施例可实施于一种流式计算系统，该系统可以包括上游计算节点以及下游计算节点，系统为各计算节点设置预定义行为程序，用于处理任务数据中相应的部分。那么相应地，在一些可行的实施方式中，通过本地的上游计算节点接收任务数据之前，本发明实施例还可以实施以下步骤：

[0068] 为本地中的各计算节点设置预定义行为程序。其中，各计算节点设置预定义行为程序可以是从系统的应用程序编程接口 API 输入的，节点的预定义行为程序也可以根据实际情况修改。

[0069] 在一些可行的实施方式中，为本地中的各计算节点设置预定义行为程序之前，通过本地的上游计算节点接收任务数据之后，本发明实施例还可以实施以下步骤：

[0070] 定义各计算节点之间的数据流向关系。

[0071] 其中，定义计算节点之间的数据流向可以由系统中的计算节点管理器来实施，可将任务数据按照任务的行为内容来划分，通过指定的计算节点处理对应的行为，并且通过定义好节点数据流向的多个计算节点所组成的数据流扑拓关系来控制数据的计算过程，可以实现将任务数据进行并行计算，通过多个计算节点之间的配合，可提高任务数据的处理效率。另外，定义节点数据流向可以依据配置文件。

[0072] 在一些可行的实施方式中，上游计算节点接收到的为多个任务数据，调用上游计算节点的预定义行为程序对任务数据进行计算，生成上游计算节点计算结果的具体实施可以包括：

[0073] 将多个任务数据插入节点输入队列；

[0074] 按照节点输入队列中任务数据的排列顺序分配多个任务数据至上游计算节点的计算单元；

[0075] 使计算单元计算分配到的任务数据，其中，上游计算节点包含至少一个计算单元；

[0076] 将至少一个计算单元计算出的计算结果插入节点输出队列，形成上游计算节点计算结果。

[0077] 其中，本发明实施例中，无论是上游计算节点还是下游计算节点以及其他计算节点均包含多个计算单元，也可以是指定数目个计算单元，每个计算单元均可以包含多个线程，也可以是指定数目的计算线程，用来快速地计算各计算节点接收的任务数据：

[0078] 为分配到计算单元的多个任务数据分配计算线程；

[0079] 使计算线程处理分配到的任务数据。

[0080] 在一些可行的实施方式中，计算线程处理任务数据则是通过将任务数据插入队列后，从队列前端抽取任务数据并处理，并将处理后的结果输出到输出队列：

[0081] 将分配到计算线程的多个任务数据插入线程输入队列；

[0082] 按照线程输入队列中任务数据的排列顺序处理数据，生成计算结果并输出到线程输出队列。

[0083] 在一些可行的实施方式中，各计算节点、节点单元以及线程的输入队列大小都是预设的，以控制数据流。

[0084] 在一些可行的实施方式中，请一并参照以下程序，本发明实施例的任务数据的流向可以如以下程序所示：

[0085] [node](node 表示一个计算节点的标识符)

[0086] node_name:node_a(计算节点的名字，下同)

[0087] input_queue_size:100(计算节点输入队列的大小，用于流量控制)

[0088] task_class:task_a(计算单元中的计算任务)

[0089] downstream:node_b,node_c(下游计算节点的名字)

[0090] [node]

[0091] node_name:node_b

[0092] input_queue_size:50

[0093] task_class:task_b

[0094] downstream:node_c

[0095] [node]

[0096] module_name:node_c

[0097] input_queue_size:10

[0098] task_class:task_c

[0099] downstream:null# 表示不存在下游

[0100] 上述程序中，节点 A 作为上游计算节点，它的数据流向所在的下游计算节点有节点 B 和节点 C，并且，节点 B 的作为上游计算节点时，节点 B 的数据流向所在的下游计算节点为节点 C，另外，节点 C 在本程序中作为终止节点，规定其不存在下游数据流向。

[0101] 在一些可行的实施方式中，系统还可以包括读代理、写代理，当计算节点接收大量任务数据后，计算节点将大量的任务数据插入计算节点输入队列，并通过节点的读代理读取计算节点输入队列中的任务数据，分配任务数据至计算单元，当计算单元通过其中的计算线程对任务数据进行处理后，将计算结果输出至内部的输出队列，再通过写代理读取内部的输出队列中的数据，并写入计算节点的输出队列，即下一个计算节点的输入队列。

[0102] 在一些可行的实施方式中，计算节点管理器用于管理节点，在处理任务数据之前，计算节点管理器要执行初始化，使每个计算节点及计算节点之间的拓扑关系初始化完成：

[0103] 首先，完成节点的读代理初始化；

[0104] 其次，完成节点的计算单元组初始化，其中，也包含了每个计算单元的初始化。每个计算单元初始化会启动任务数据。

[0105] 再次，完成节点的写代理初始化（可选，当计算节点中的任务数据处理后无需输出时，计算节点的输出队列可置为 null，写代理组件则不需要）。

[0106] 在本发明实施例中，为各节点定义了处理任务数据的行为程序，并配置了各节点的输入队列大小以及节点之间的上下游关系，从而将处理任务数据的多项行为分配到本地的各个计算节点，使各计算节点调用各节点的预定义行为程序并行、协同地处理任务数据，提高了计算效率，避免了程序资源浪费，实现了轻量化计算。

[0107] 参见图 2，是本发明实施例提供的流式计算系统的实施例结构示意图。本实施例中所描述的流式计算系统，包括：

[0108] 上游计算节点 21，位于本地，用于接收任务数据；还用于调用上游计算节点的预定义行为程序对任务数据进行计算，生成上游计算节点计算结果；还用于发送上游计算节点计算结果至位于上游计算节点数据流向所在的下游计算节点；

[0109] 下游计算节点 22，为上游计算节点的数据流向所在，用于接收上游计算节点发送的上游计算节点计算结果；还用于调用下游计算节点的预定义行为程序对上游计算节点计算结果进行计算，生成下游计算节点计算结果；还用于继续发送下游计算节点计算结果至位于下游计算节点数据流向所在的下一下游计算节点，使下一下游节点调用下一下游节点的预定义行为程序对下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出任务数据的最终计算结果。

[0110] 具体实现中，本发明实施例的系统选取了本地多个节点来计算任务数据，每个节点均配置有预定义行为程序，用于处理任务数据。

[0111] 在一些可行的实施方式中，本发明实施例的系统还可以包括：

[0112] 计算节点管理器 23，用于为本地中的各计算节点设置预定义行为程序。其中，各节点设置预定义行为程序可以是从系统的应用程序编程接口 API 输入的，节点的预定义行为程序也可以根据实际情况修改。

[0113] 在一些可行的实施方式中，计算节点管理器 23，还用于定义各计算节点之间的数据流向关系。其中，定义计算节点之间的数据流向可以由系统中的计算节点管理器 23 来实施，可将任务数据按照任务的行为内容来划分，通过指定的计算节点处理对应的行为，并且通过定义好节点数据流向的多个计算节点所组成的数据流扑拓关系来控制数据的计算过程，可以实现将任务数据进行并行计算，通过多个计算节点之间的配合，可提高任务数据的处理效率。另外，定义节点数据流向可以依据配置文件。

[0114] 在一些可行的实施方式中，上游计算节点接收到的为多个任务数据时，上游计算节点 21，还用于将多个任务数据插入节点输入队列；还用于按照节点输入队列中任务数据的排列顺序分配多个任务数据至上游计算节点的计算单元 24；

[0115] 上游计算节点 21 可以进一步包括：

[0116] 计算单元 24，用于计算任务数据，上游计算节点包含至少一个计算单元；计算单元 24，还用于将至少一个计算单元计算出的计算结果插入节点输出队列，形成上游计算节点计算结果。

[0117] 其中，本发明实施例中，无论是上游计算节点21还是下游计算节点22以及其他计算节点均包含多个计算单元24，也可以是指定数目个计算单元24，每个计算单元24均可以包含多个计算线程25，也可以是指定数目的计算线程25，用来快速地计算各计算节点接收的任务数据：

[0118] 计算单元，还用于为分配到计算单元的为多个任务数据分配计算线程；

[0119] 计算线程25，用于处理计算单元分配的任务数据。

[0120] 在一些可行的实施方式中，计算线程25处理任务数据则是通过将任务数据插入队列后，从队列前端抽取任务数据并处理，并将处理后的结果输出到输出队列：

[0121] 计算线程25，还用于将分配到计算线程的多个任务数据插入线程输入队列；还用于按照线程输入队列中任务数据的排列顺序处理数据，生成计算结果并输出到线程输出队列。

[0122] 在一些可行的实施方式中，系统中各节点、节点单元以及线程的输入队列大小都是预设的，以控制数据流。

[0123] 在一些可行的实施方式中，请一并参照以下程序，本发明实施例的系统中，任务数据的流向可以如以下程序所示：

[0124] [node](node表示一个计算节点的标识符)

[0125] node_name:node_a(计算节点的名字，下同)

[0126] input_queue_size:100(计算节点输入队列的大小，用于流量控制)

[0127] task_class:task_a(计算单元中的计算任务)

[0128] downstream:node_b,node_c(下游计算节点的名字)

[0129] [node]

[0130] node_name:node_b

[0131] input_queue_size:50

[0132] task_class:task_b

[0133] downstream:node_c

[0134] [node]

[0135] module_name:node_c

[0136] input_queue_size:10

[0137] task_class:task_c

[0138] downstream:null# 表示不存在下游

[0139] 上述程序中，系统中的节点A作为上游计算节点，它的数据流向所在的下游计算节点有节点B和节点C，并且，节点B的作为上游计算节点时，节点B的数据流向所在的下游计算节点为节点C，另外，节点C在本程序中作为终止节点，规定其不存在下游数据流向。

[0140] 在一些可行的实施方式中，系统还可以包括读代理、写代理，当计算节点接收大量任务数据后，计算节点将大量的任务数据插入计算节点输入队列，并通过计算节点的读代理读取计算节点输入队列中的任务数据，分配任务数据至计算单元，当计算单元通过其中的计算线程处理任务数据后，将计算结果输出至内部的输出队列，再通过写代理读取内部的输出队列中的数据，并写入计算节点的输出队列，即下一个计算节点的输入队列。

[0141] 在一些可行的实施方式中，计算节点管理器用于管理节点，在处理任务数据之前，

计算节点管理器要执行初始化，使每个计算节点及计算节点之间的拓扑关系初始化完成：

[0142] 首先，完成节点的读代理初始化；

[0143] 其次，完成节点的计算单元组初始化，其中，也包含了每个计算单元的初始化。每个计算单元初始化会启动任务数据。

[0144] 再次，完成节点的写代理初始化（可选，当计算节点中的任务数据处理后无需输出时，计算节点的输出队列可置为 null，写代理组件则不需要）。

[0145] 在本发明实施例中，为各节点定义了处理任务数据的行为程序，并配置了各节点的输入队列大小以及节点之间的上下游关系，从而将处理任务数据的多项行为分配到本地的各个计算节点，使各计算节点调用各节点的预定义行为程序并行、协同地处理任务数据，提高了计算效率，避免了程序资源浪费，实现了轻量化计算。

[0146] 在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中，对上述术语的示意性表述不必须针对的是相同的实施例或示例。而且，描述的具体特征、结构、材料或者特点可以在任一个或多个实施例或示例中以合适的方式结合。此外，在不相互矛盾的情况下，本领域的技术人员可以将本说明书中描述的不同实施例或示例以及不同实施例或示例的特征进行结合和组合。

[0147] 此外，术语“第一”、“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。在本发明的描述中，“多个”的含义是至少两个，例如两个，三个等，除非另有明确具体的限定。

[0148] 流程图中或在此以其他方式描述的任何过程或方法描述可以被理解为，表示包括一个或更多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分，并且本发明的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能，这应被本发明的实施例所属技术领域的技术人员所理解。

[0149] 在流程图中表示或在此以其他方式描述的逻辑和 / 或步骤，例如，可以被认为是用于实现逻辑功能的可执行指令的定序列表，可以具体实现在任何计算机可读介质中，以供指令执行系统、装置或设备（如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统）使用，或结合这些指令执行系统、装置或设备而使用。就本说明书而言，"计算机可读介质"可以是任何可以包含、存储、通信、传播或传输程序以供指令执行系统、装置或设备或结合这些指令执行系统、装置或设备而使用的装置。计算机可读介质的更具体的示例（非穷尽性列表）包括以下：具有一个或多个布线的电连接部（电子装置），便携式计算机盘盒（磁装置），随机存取存储器 (RAM)，只读存储器 (ROM)，可擦除可编辑只读存储器 (EPROM 或闪速存储器)，光纤装置，以及便携式光盘只读存储器 (CDROM)。另外，计算机可读介质甚至可以是可在其上打印所述程序的纸或其他合适的介质，因为可以例如通过对纸或其他介质进行光学扫描，接着进行编辑、解译或必要时以其他合适方式进行处理来以电子方式获得所述程序，然后将其存储在计算机存储器中。

[0150] 应当理解，本发明的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中，多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。例如，如果用硬件来实现，和在另一实施方式中一样，可用本领域公知的下列技术中的任一项或他们的组合来实现：具有用于对数据信号实现逻辑功能的逻辑门电路的离散逻辑电路，具有合适的组合逻辑门电路的专用集成电路，可编程门阵列 (PGA)，现场可编程门阵列 (FPGA) 等。

[0151] 本技术领域的普通技术人员可以理解实现上述实施例方法携带的全部或部分步骤是可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，该程序在执行时，包括方法实施例的步骤之一或其组合。

[0152] 此外，在本发明各个实施例中的各功能单元可以集成在一个处理模块中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个模块中。上述集成的模块既可以采用硬件的形式实现，也可以采用软件功能模块的形式实现。所述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用时，也可以存储在一个计算机可读取存储介质中。

[0153] 上述提到的存储介质可以是只读存储器，磁盘或光盘等。尽管上面已经示出和描述了本发明的实施例，可以理解的是，上述实施例是示例性的，不能理解为对本发明的限制，本领域的普通技术人员在本发明的范围内可以对上述实施例进行变化、修改、替换和变型。

[0154] 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。

[0155] 以上所揭露的仅为本发明较佳实施例而已，当然不能以此来限定本发明之权利范围，因此依本发明权利要求所作的等同变化，仍属本发明所涵盖的范围。

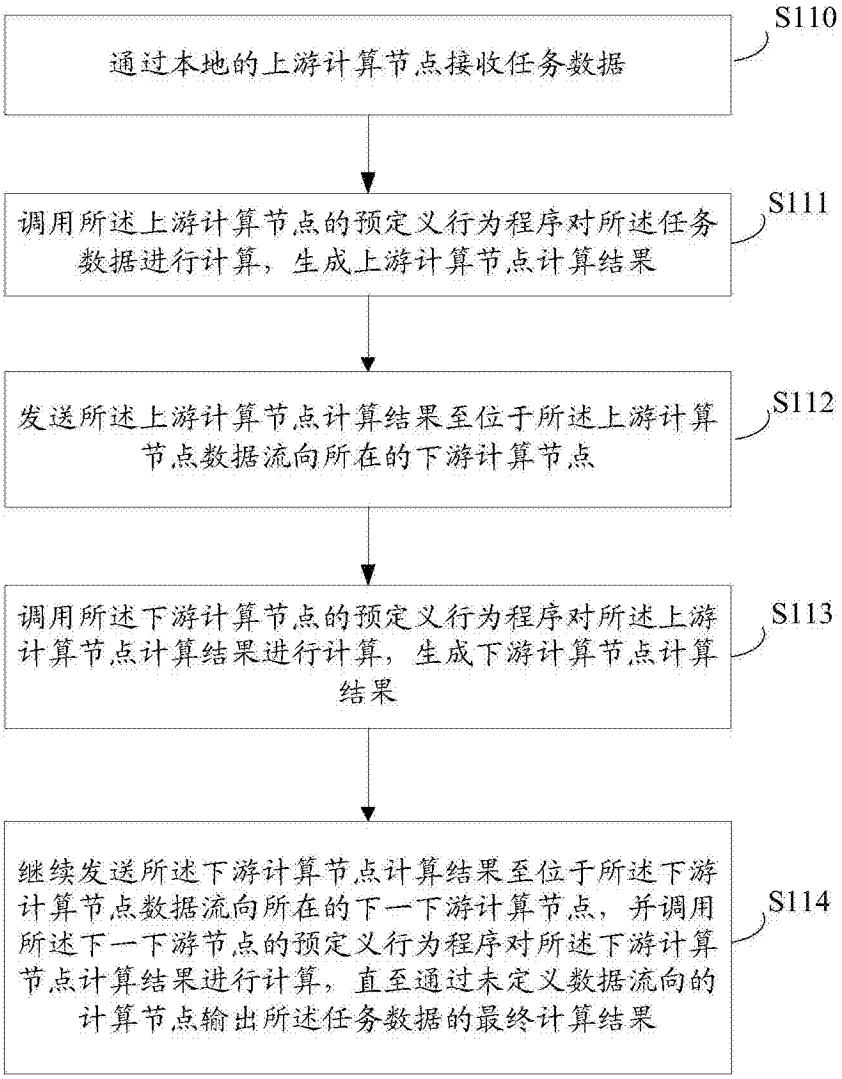

S110
通过本地的上游计算节点接收任务数据
S111
调用所述上游计算节点的预定义行为程序对所述任务数据进行计算，生成上游计算节点计算结果
S112
发送所述上游计算节点计算结果至位于所述上游计算节点数据流向所在的下游计算节点
S113
调用所述下游计算节点的预定义行为程序对所述上游计算节点计算结果进行计算，生成下游计算节点计算结果
S114
继续发送所述下游计算节点计算结果至位于所述下游计算节点数据流向所在的下一下游计算节点，并调用所述下一下游节点的预定义行为程序对所述下游计算节点计算结果进行计算，直至通过未定义数据流向的计算节点输出所述任务数据的最终计算结果

图 1

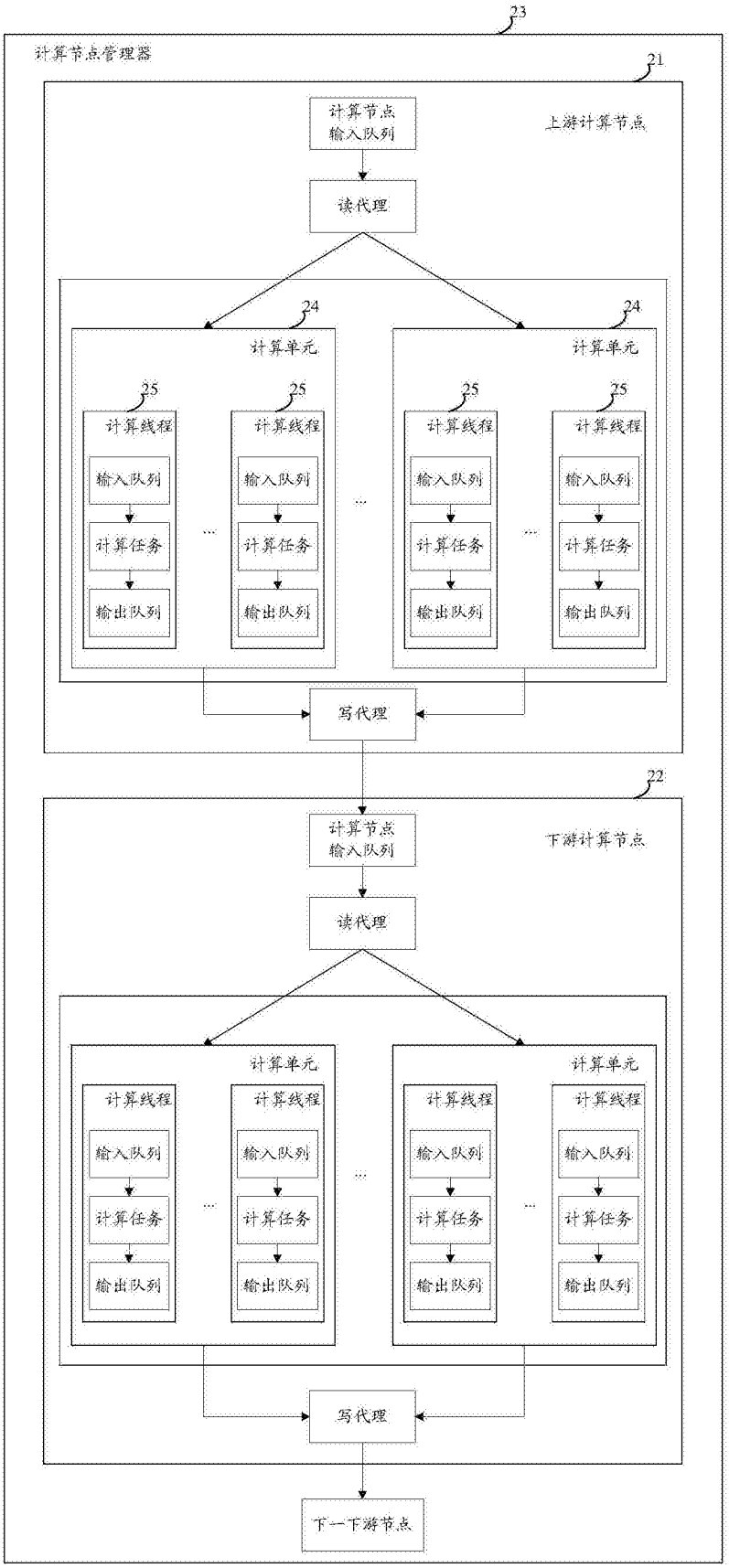

23
计算节点管理器
21
计算节点
输入队列
上游计算节点
读代理
24
计算单元
25
计算线程
输入队列
计算任务
输出队列
写代理
22
下游计算节点
下一下游节点

图 2