

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/46 (2006.01)



[12] 发明专利说明书

专利号 ZL 200580007518.7

[45] 授权公告日 2008 年 10 月 15 日

[11] 授权公告号 CN 100426239C

[22] 申请日 2005.3.8

[21] 申请号 200580007518.7

[30] 优先权

[32] 2004.3.8 [33] US [31] 10/795,374

[86] 国际申请 PCT/US2005/007923 2005.3.8

[87] 国际公布 WO2005/086906 英 2005.9.22

[85] 进入国家阶段日期 2006.9.8

[73] 专利权人 AB 开元软件公司

地址 美国马萨诸塞州

[72] 发明人 弗兰克·因金戈洛

克雷格·W.·斯坦菲尔

[56] 参考文献

US6263358B1 2001.7.17

US6490611B1 2002.12.3

US6675189B2 2004.1.6

US6480876B2 2002.11.12

US2003/0149714A1 2003.8.7

JAPS 中任务调度协议的设计与实现. 李
晓明, 陈道蓄, 谢立. 计算机科学, 第 27 卷第
12 期. 2000

审查员 李 俊

[74] 专利代理机构 隆天国际知识产权代理有限公司

代理人 王玉双

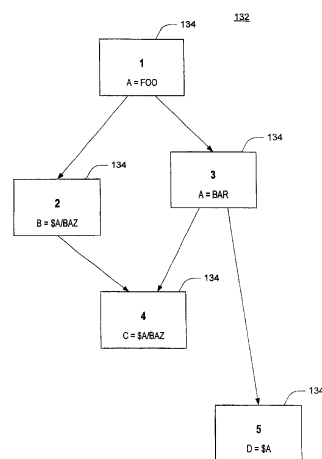
权利要求书 3 页 说明书 10 页 附图 9 页

[54] 发明名称

依赖关系图参数范围的划分

[57] 摘要

根据依赖关系图定义多个任务。保持分别与不同的任务范围相关的多个参数范围。在第一任务中使用的参数与参数值绑定。这种绑定操作包含根据依赖关系图识别第一范围, 并根据识别的范围来检索所述参数值。



1. 一种用于绑定参数的值的方法，包括：

根据依赖关系图定义多个任务，其中所述依赖关系图表示所述任务的部分定序；

保持分别与不同任务范围相关的多个参数范围；

在第一参数范围中存储一个或多个参数值；以及

将参数值与第一任务中使用的第一参数绑定，所述绑定操作包括根据所述依赖关系图所表示的所述部分定序识别所述第一参数范围，并根据所识别的范围检索所述第一参数的值。

2. 根据权利要求 1 所述的方法，其中在第一参数范围中存储一个或多个参数值包括：

在第二任务执行期间确定赋予所述第一参数的值；以及

将所确定的赋予所述第一参数的值存储到所述第一参数范围中。

3. 根据权利要求 2 所述的方法，其中所述第一参数范围与所述第二任务相关。

4. 根据权利要求 1 所述的方法，其中所述依赖关系图指定所述任务的执行顺序，并且其中识别第一参数范围的操作包括识别必须在所述第一任务之前执行的任务。

5. 根据权利要求 4 所述的方法，其中识别所述第一参数范围的操作包括识别对所述第一参数赋值的任务。

6. 根据权利要求 5 所述的方法，其中识别对所述第一参数赋值的任务的操作包括根据所述依赖关系图识别最后执行的任务。

7. 根据权利要求 1 所述的方法，其中给定参数的多个实例分别与各种不同的参数范围相关。

8. 根据权利要求 7 所述的方法，其中所述给定参数的多个实例分别根据不同的任务来赋值。

9. 一种任务管理系统，包括：

任务管理器定义模块，其被配置为根据依赖关系图定义多个任务，其中所述依赖关系图表示所述任务的部分定序；以及

任务管理器动态范围模块，其具有存储空间，用以保持分别与不同任务

范围相关的多个参数范围；以及

任务管理器参数绑定模块，其能够访问所述参数范围，并被配置为在第一参数范围中存储一个或多个参数值以及将参数值与第一任务中使用的的第一参数绑定，所述绑定操作包括根据所述依赖关系图所表示的所述部分定序识别所述第一参数范围，并根据所识别的范围检索所述第一参数的值。

10. 根据权利要求 9 所述的系统，其中在第一参数范围中存储一个或多个参数值包括：

在第二任务执行期间确定赋予所述第一参数的值；以及

将所确定的赋予所述第一参数的值存储到所述第一参数范围中。

11. 根据权利要求 10 所述的系统，其中所述第一参数范围与所述第二任务相关。

12. 根据权利要求 9 所述的系统，其中所述依赖关系图指定所述任务的执行顺序，并且其中识别第一参数范围的操作包括识别必须在所述第一任务之前执行的任务。

13. 根据权利要求 12 所述的系统，其中识别所述第一参数范围的操作包括识别对所述第一参数赋值的任务。

14. 根据权利要求 13 所述的系统，其中识别对所述第一参数赋值的任务的操作包括根据所述依赖关系图识别最后执行的任务。

15. 一种任务管理系统，包括：

用于根据依赖关系图定义多个任务的装置，其中所述依赖关系图表示所述任务的部分定序；

用于保持分别与不同任务范围相关的多个参数范围的装置；

用于在第一参数范围中存储一个或多个参数值的装置；以及

用于将参数值与第一任务中使用的的第一参数绑定的装置，所述绑定操作包括根据所述依赖关系图所表示的所述部分定序识别所述第一参数范围，并根据所识别的范围检索所述第一参数的值。

16. 根据权利要求 15 所述的系统，其中在第一参数范围中存储一个或多个参数值包括：

在第二任务执行期间确定赋予所述第一参数的值；以及

将所确定的赋予所述第一参数的值存储到所述第一参数范围中。

17. 根据权利要求 16 所述的系统, 其中所述第一参数范围与所述第二任务相关。

18. 根据权利要求 15 所述的系统, 其中所述依赖关系图指定所述任务的执行顺序, 并且其中识别第一参数范围的操作包括识别必须在所述第一任务之前执行的任务。

19. 根据权利要求 18 所述的系统, 其中识别所述第一参数范围的操作包括识别对所述第一参数赋值的任务。

20. 根据权利要求 19 所述的系统, 其中识别对所述第一参数赋值的任务的操作包括根据所述依赖关系图识别最后执行的任务。

依赖关系图参数范围的划分

技术领域

本发明涉及依赖关系图中的参数范围的划分。

背景技术

计算机作业控制系统已经被用于根据与待执行程序的定序相关的限制来顺序执行计算机程序或其它处理。一种指定定序限制的方法是使用依赖关系图。程序可接受在将作业提交给系统时所指定的变量。这类作业控制系统的一个示例是分布式批处理系统，其中可指定不同的作业在不同的计算机或不同类型的计算机（例如超级计算机）上执行，并且不同作业之间的依赖关系在作业被执行之前清楚地标识。一种在作业之间传送信息的方法是通过公用数据存储，例如通过文件系统中的文件进行。这类作业控制的另一示例涉及分布式个人计算机环境中的作业调度，例如，调度需要以特定顺序在不同计算机上执行的维护任务。

发明内容

作为概括方案，本发明提供一种用于绑定参数值的方法。根据依赖关系图定义多个任务，其中所述依赖关系图表示所述任务的部分定序。保持分别与不同的任务范围关联的多个参数范围。在第一参数范围中存储一个或多个参数值。在第一任务中使用的参数与参数值绑定。这种绑定操作包括根据依赖关系图所表示的所述部分定序识别所述第一参数范围，并根据所识别的范围检索所述第一参数的值。

这一方案可包括一个或多个以下特征：

在第二任务执行期间确定赋予参数的值。所确定的赋予所述参数的值被存储在与第二任务相关的第一参数范围中。

所述依赖关系图指定任务的执行顺序。识别所述第一参数范围的操作包括识别必须在第一任务之前执行的任务。识别所述第一参数范围的操作还包

括识别对所述第一参数赋值的任务，并且根据所述依赖关系图识别最后执行的任务。

作为另一概括方案，本发明还提供一种用于检查参数访问的方法，该方法包括接受对多个任务的定序限制的描述内容，其中至少一些任务未彼此定序。这种检查操作包括：对一个或多个参数中的每个参数，识别访问这些参数的任务；以及根据定序限制的描述内容，检查通过识别的任务对参数的访问中潜在的冲突。

本发明的方案可包含一个或多个下列优点：

利用在各任务之间传送的显式参数，能够检查参数值中的冲突和模糊性。例如，这种检查能够确保各参数在任务计划的重复执行中将取相同的值。

根据依赖关系图中的优先级的范围划分优选为根据嵌套或计划和子计划进行的范围划分，这是因为参数可在计划中的同级任务之间传送。

通过允许在依赖关系图的不同部分中使用的相同参数名能够根据范围划分规则被多重使用而不引入冲突或模糊性，多种参数范围的使用可减少名称冲突。这样就能够允许不同的开发者在不引入名称冲突并且不必清楚标识不同的命名域（naming domain）的情况下编写计划的不同部分。

根据以下描述和权利要求书，本发明的其它特征和优点是显而易见的。

附图说明

图 1 为任务管理系统。

图 2 为依赖关系图。

图 3 为包含任务管理器中的单个全局范围（global context）的示图。

图 4A 为包含任务管理器中的基于特定任务的多个动态范围的框图。

图 4B 为根据图 2 的依赖关系图展示各动态范围之间的关系关系的示图。

图 4C 为展示图 2 的依赖关系图中的任务范围的示图。

图 5 为任务描述内容。

图 6A-B 为使用动态参数的应用示例。

具体实施方式

概述

参照图 1，任务管理系统 100 根据依赖性限制控制构成整个计划的任务的执行，所述依赖性限制指定计划中的哪些任务必须在执行其它任务之前完成。依赖性限制是用定向依赖关系图 132 来表示的，在依赖关系图 132 中该图的每一节点 134 对应于计划中的不同任务，而从一个节点到另一节点的每个定向弧 136 表示这样的定序限制：对应于第一个（“上游”）节点的任务必须在执行对应于第二个（“下游”）节点的任务之前完成。更一般地说，对应于节点的任务仅可以在所述图中其所有上游任务均已完成之后再执行。所述依赖关系图建立了任务的部分定序。因此，任务可按照限制而同时执行，这些限制包括基于图的依赖性限制。任务也可在不同的计算机上执行，以在整个计划执行中实现并行性加速。

任务管理系统 100 包括任务管理器 110，该管理器读取计划描述内容 120。上述计划描述内容包括定义依赖关系图 132 的依赖关系图描述内容 122 及定义每个任务特性的任务描述内容 124。依赖关系图描述内容 122 通过一系列节点 134 和一系列弧 136 来定义依赖关系图，每个弧 136 标识该图中的源节点和目标节点。每一节点与任务定义相关，该任务定义被指定于任务描述内容 124 中，任务描述内容 124 指定该任务中所要执行的特定操作。

任务可用任务模板来定义，而计划可用计划模板来定义。任务或计划的模板可包括对参数的引用，所述对参数的引用是以符号表示的，并且在任务被执行前不必与值绑定。例如，任务模板可包括对被命名为 FILENAME 的参数的引用，并且对该参数的引用利用“美元符号”语法 \$FILENAME 来表示并在执行任务之前或执行任务期间通过任务管理器 110 与特定值绑定。

有些参数对整个计划来说是全局性的，并且在整个计划第一次执行时被赋值。这类参数总体上可以是计划的形式参数，并可以，举例来说，在调用总体计划的命令中加以指定，或可由调用计划的用户以交互方式引出。

其它参数无需在计划第一次执行时指定值。这类动态参数可在一个或多个任务的执行期间或作为执行结果而被赋值，并随后为其它任务的模板所引用。例如在图 2 中，参数 A 可通过任务 1 而被设定成值，并且该值可随后由任务 2 使用。

任务管理系统 100 支持多个与以下讨论有关的附加特征。第一个特征是，节点 134 可与整个“子计划”相关，而非与单个任务相关。子计划是部分定

序任务的较低级计划。子计划具有与上级计划描述内容相同或相似的结构计划描述内容 120。

第二个特征是任务能够通过其执行来修改计划描述内容 120。特别是，任务能够将附加节点 134 和弧 136 添加到依赖关系图 132，并且利用由任务管理器 110 所提供的服务来定义与所添加的节点相关的任务的特性。例如，对于具有未知（在运行之前）数目的待完成任务的计划，第一任务可利用模板产生子计划。使用动态参数在运行时绑定模板中任务的数目。所产生的子计划由任务管理器 110 合并到第一任务的较高级计划。

绑定方法

如下所述，任务管理器 110 利用多个可选方法之一进行参数绑定。或者，任务管理器 110 利用多种绑定方法的结合，例如，根据这些参数的声明而对不同的参数采用不同的方法。其中部分（或全部）方法不同的一个方面在于参数定义的范围。

方法 1

参照图 3，在第一绑定方法中，任务管理器 110 对所有参数保持单一全局范围 310。任务管理器 110 保持任何参数的单一拷贝。也就是说，它保持数据结构 320，该数据结构使得多个不同的参数名 322 均与该参数的当前值 324 相关联（其中若相应参数尚未被赋值则值 324 可以为空）。此相同数据结构 320 可被用来保存全局参数的值以及那些随着任务的执行通过任务而被动态赋予的值。

参照图 2 和图 3，在上述第一方法的示例中，任务 1 对参数 A 赋予值 FOO，如赋值语句 $A=FOO$ 所示。任务管理器 110 从任务 1 接收所赋的值 FOO（从任务 1 到参数 A 的值单元 324 的箭头表示），并将该值存储于值单元 324 内。任务管理器 110 此后将值 FOO 提供给任务 2，以便在任务 2 的模板中绑定对参数 A 的引用。在这一示例中，任务 2 包含赋值语句 $B=\$A/BAZ$ ，这能够使得参数 B 被赋值 FOO/BAZ。任务管理器 110 接收存储在全局范围 310 中的 B 的赋值。根据依赖关系图 132，任务 2 仅在任务 1 完成之后执行。因此，对参数 A 的赋值 FOO 保证在执行任务 2 之前即已执行。

通常，可同时执行计划的多项任务组。针对特定依赖关系图，如果在依赖关系图中未彼此定序的两个或更多个任务之间没有直接路径，则这些任务

可能被同时执行，或者以不可预知的顺序执行。在图 2 中，任务 2 及 3 便是这样的未定序任务。在图 2 所示示例中，任务 3 对参数 A 赋值 BAR。如果任务 3 在任务 2 之前执行，则参数 B 在任务 2 中被赋值 BAR/BAZ，而如果任务 3 在任务 2 之后执行，参数 B 即被赋值 FOO/BAZ。

在许多应用中，关于哪个值将被赋予 B 的不确定性是不希望的。避免这种情况的一种方法是执行依赖关系图的静态检查，并且禁止使用表现出这种不确定性的计划。所述检查包含识别未被依赖关系图定序的各任务组。如果特定参数是由该组中一个任务使用而由该组中另一任务赋值的，则产生潜在的模糊性。类似地，如果在这样的组中的两个任务对一特定参数赋予多个值，也会产生冲突。通常，在其中多个任务访问一特定参数的未定序任务的任何组中，没有任务能给该参数赋值。这种静态检查会根据任务 2 与 3 对参数 A 的引用识别它们之间的冲突。任务管理系统 100 处理由静态检查所识别的冲突（例如利用错误处理机制）。注意在本示例中，任务 2 及 5 也可能以不可预知的顺序执行，但二者均使用参数 A 的值而非给参数 A 赋值，因此并不表现出冲突或模糊性。

依赖关系图的静态检查是在计划执行之前由任务管理器 110 执行的，或可选择地由确认计划描述内容的另一模块以预处理的方式来执行。在通过任务的执行来修改依赖关系图的情况下，再次执行或增加静态检查，以用于依赖关系图的修改。

对于包括子计划的计划，可采用上述方法的两种变形。在第一种变形中，任务管理器 110 对于所有级别的子计划的所有参数保持单个范围。在对包含具有子计划的节点的计划进行的静态检查中，由该子计划的任何任务所进行的参数赋值与使用该参数值的较高级计划中的任何未定序任务（对于具有该子计划的节点而言）冲突。

在涉及子计划的第二种变形中，任务管理器 110 对每个子计划的参数保持单独的范围，并且子计划的描述内容明确地标识当子计划完成时从子计划输出到下一个较高级计划的参数。在这类变形中，静态检查涉及子计划的输出参数与在下一个较高级计划内的任务（该任务不依子计划的执行来定序）中对该参数的访问之间的冲突。

在这种方法的另一版式中，任务管理器 110 禁止对任何特定全局动态参

数的重复赋值。

方法2

参照图 4A，在第二方法中，任务管理器 110 对于为整个计划设置的参数保持一全局范围 310，以及保持分别与依赖关系图的不同节点相关的多个动态范围 410。对参数赋值的每个任务产生该参数的新“实例（instance）”并将赋值（通过任务管理器 110）存储在与对应于该任务的节点相关的动态范围 410 中。任务管理器 110 因此可存储与参数相关的多个值，每个值与该参数的不同实例相关。对于任何特定参数（例如 A），任务管理器 110 保持与其中参数被赋值的每个任务相关的参数的单独值（因为每个任务对应于单独节点和单独动态范围 410）。

参照图 4B 及 4C，参数的每个实例及其相关动态范围 410 具有确定哪个任务“得知”对该实例赋予的值的任务范围。根据依赖关系图 132 的图形 420，每个动态范围 410 基于各动态范围 410 之间的关系与不同的任务范围相关。例如，在任务 1 中被赋值为 A=FOO 的参数 A 的实例（存储在动态范围 410 中）具有任务范围 430，该任务范围不同于任务范围 432，所述任务范围 432 与在任务 3 中被赋值为 A=BAR 的参数 A 的实例相关。与参数 B 的实例相关的任务范围 434 包含任务 4 但不包含任务 5，所述参数 B 在任务 2 中被赋值为 B=\$A/BAZ。在这种方法中，任务范围或参数实例的“范围”（及其相关动态范围 410）仅包含将该值赋予参数实例的任务以及在该任务之后（“下游的”）定序的任务。当任务引用参数时，任务管理器 110 将其范围包括该任务的参数实例的值绑定到该引用。

当任务 1 赋值 FOO 给参数 A 时，任务管理器 110 将该值存储在与任务 1 相关的动态范围 410 中。类似地，当任务 3 赋值 BAR 给参数 A 时，任务管理器 110 将该值存储在与任务 3 相关的动态范围 410 中。任务 2 处于在任务 1 中被赋值成 A=FOO 的参数 A 的实例的范围内，但不处于在任务 3 中被赋值成 A=BAR 的参数 A 的实例的范围内。因此，在这种范围划分方法下，对于任务 2 中的 B 值不存在不确定性或模糊性。任务管理器 110 将任务 2 中对参数 A 的引用与来自任务 1 的动态范围 410 的值绑定。任务 2 将 B 赋值为 FOO/BAZ，该值提供给任务管理器 110 从而存储在与任务 2 相关的动态范围 410 中。

通常在执行期间，任务管理器 110 通过考虑经过依赖关系图的上游路径（即经过单个弧的直接路径或经过已执行任务的非直接路径），使值与对任务中参数的引用绑定，其中所述路径以对该参数赋值的任务终止。在此示例中，当绑定任务 2 中的参数 A 时，任务管理器 110 采用已设定参数的任务 1 的上游路径。

在图 2 的示例中，任务 4 潜在地处于任务 1 参数 A 的实例范围之内、以及处于任务 3 的参数 A 的实例范围之内。也就是说，存在从任务 4 到达任务 1（非直接而是经任务 2）和到达任务 3（直接地）的上游路径，任务 1 和任务 3 各自给 A 赋值。然而在本示例中不存在模糊性，因为任务 3 被限制成在任务 1 之后执行。如果有多个上游参数赋值，且如果有一个任务被定序到所有其它任务之后，则不存在模糊性，并且使用赋予最后任务的动态范围 410 中参数实例的值。

与第一种方法相同，潜在的模糊性可根据依赖关系图静态地进行检查。具体地说，对于依赖关系图中节点（任务）处对参数的每个引用，首先识别可给该参数赋值的上游任务。如果没有这样的任务，则此参数将会具有未定义值或从计划的全局范围提供该参数。如果确实有一个上游任务，则不存在模糊性。如果有多个对该参数赋值的上游任务，则存在模糊性，除非这些任务中有单独一个被定序到其它任务之后（即通过依赖关系图被限制在其它任务之后执行）。

实施方法

参照图 5，任务描述内容 500（其包含在图 1 中的任务描述内容 124 内）包含声明部分 510 和指令部分 520。声明部分 510 标识（例如用一个或多个声明语句）通过相应任务的执行而被赋值的动态参数。图 5 中所示任务描述内容对应于图 2 中的任务 1。参数 A 被声明为由任务来动态赋值，表示成声明“dynamicA”512。声明部分 510 还标识对需要由任务管理器 110 根据可用参数的范围来绑定的参数的引用。

指令部分 520 包含待由任务来执行的操作的计算机指令（例如编程语言或其它过程语句），以及对动态参数赋值的指令，例如指令“A=FOO”522。这类计算机指令，举例来说，可由用户指定或通过计算机处理而自动产生。

任务管理器 110 能够处理计划的任务描述内容的声明部分，以确定在相

应任务的执行期间要被赋值、或引用、或者既赋值又引用的参数。任务管理器 110 可通过额外地处理任务描述内容的指令部分，或可选择地通过使用单独类型的用于动态参数赋值的声明语句或用于引用的声明语句，在赋值或引用之间进行区分。例如，对于图 2 中所示的示例，任务管理器 110 可解析任务描述内容，从而确定任务 2 引用参数 A 而任务 3 对 A 赋值。这使得任务管理器 110 能够完成对第一种方法（见图 3，其中任务 2 中的引用会与任务 3 中的赋值冲突）的静态检查。

当声明部分标识待引用的参数时，任务管理器 110 能够在执行指令部分 520 的各指令之前配置参数及其绑定值的环境。在配置了任务的环境后，任务管理器 110 调用执行任务描述内容的指令部分的指令的处理。该处理随后访问来自己为参数值创建的环境中的那些参数值。在执行期间，通过动态参数的赋值指令的处理，将（参数，值）对记录在任务执行的输出中，即动态参数赋值部分 530 中。例如，对（A，FOO）532 将会记录在赋值部分 530 中。任务管理器 110 在指令部分 520 的指令执行之后接收赋值部分 530，并使用其内容来更新参数值。

在此方法的一个特定实施方式中，指令部分包含 SHELL 脚本语言的语句，例如“ksh”。在调用 SHELL 处理以翻译指令之前，任务管理器 110 利用声明部分 510 来配置那些将在 SHELL 指令的处理中使用的参数值的环境。在这一实施方式中，赋值语句采用脚本的文本输出的输出语句的形式。也就是说，脚本指令将会包括诸如“PRINT A=FOO”之类语句而不采用赋值“A=FOO”，从而输出行“A=FOO”。脚本的文本输出构成动态参数赋值部分 530。任务管理器 110 接收该文本输出，并翻译将值存储在适当的动态范围 410 中的赋值语句。

在另一实施方式中，指令部分 520 中翻译指令的处理与任务管理器 110 通信，并且任务管理器 110 为存储或访问动态参数值的请求提供服务。当参数引用需要与值绑定以执行指令时，处理向任务管理器 110 发出请求，该管理器确定用于该参数的适当动态范围 410 并返回该动态范围 410 中的值。类似地，当参数被赋值时，处理发出请求以将该值存储在适当的动态范围 410 中。

在一可选的实施方式中，任务描述内容 500 的声明部分 510 不标识在执

行期间任务将会需要的参数，并且在任务的指令正在执行时无法用适当的通信机制来请求参数值。在这一实施方式中，任务管理器 110 配置其范围包含该任务的所有参数实例的完整列表。对于其绑定会导致潜在的模糊性的参数（例如因为该参数是由多个上游未定序任务赋值的），以指示符如“ambiguous_value”来取代参数值，因此若在任务中引用该参数就会产生错误。

当声明部分 510 不包含将在每个任务中被动态赋值和/或被引用的参数的声明时，依赖关系图的静态检查将是不可能的。然而，任务管理器 110 仍然能够识别在计划的执行期间可能发生的某些潜在冲突。例如，使用对单个全局范围进行范围划分的方法（见图 3），如果任务管理器 110 在计划执行期间检测到多个未定序任务已对参数赋值，则任务管理器 110 可识别潜在的冲突。任务管理系统 100 可随后处理任务管理器所识别的冲突。

示例

参照图 6A，使用动态赋值参数的示例包括通过整个计划中的不同任务 631-633 分别对多个数据文件 641-643 的处理。数据文件 641-643 的文件名在计划被第一次执行时是未知的。更确切地说，文件名被列于文本文件 622 中，其名称是作为整个计划的全局参数 LIST_FILE 提供的。

计划包括第一任务 620，该任务负责打开 LIST_FILE 文件 622，并读取其内容。此任务 620 打开文件并读取内容，确定列出的单个数据文件的数目。在这一示例中，列表文件 622 有三个文件名：FNAME1.DAT、FNAME2.DAT、及 FNAME3.DAT。任务例如利用数据网络上的文件传输协议来检索所列文件 641-643。任务 1 对多个动态参数 NUM_INPUT、及 DATA_FILE_1 到 DATA_FILE_3 赋值，并将这一赋值信息传送给任务管理器 110，该管理器将其记录到适当的范围 610（例如全局范围、或与任务 1 相关的动态范围，这取决于所用范围划分方法）中。

在示例的这个第一版本中，最初规定的计划确切地具有用于处理输入文件的任务的三个实例，即任务 2-4（631-633）。每个任务通过引用由任务 1 赋值的动态参数来检索其相应的输入数据文件名。

这个示例的变形利用了能够修改计划描述内容的任务。计划最初仅有单一任务，“获取名称”任务 1（620）。在“获取名称”任务已确定存在三个

待处理数据文件之后，该任务利用任务管理器 110 的服务来创建三个处理任务 631-633，并将它们连接到其自身（任务 1）。然后在任务 1 完成之后，如上所述，任务管理器 110 执行新创建的任务 631-633。

或者，在这个示例的另一变形中，任务管理器 110 可采用具有可修改数目任务的计划模板。参照图 6B，计划最初有两个任务：“获取名称”任务 1（620）、及“处理”任务 2（650）。在“获取名称”任务已确定存在三个待处理数据文件之后，“处理”任务利用具有可修改数目任务的计划模板创建子计划。“处理”任务引用动态参数 NUM_INPUT 来创建具有三个（未定序）处理任务 631-633 的子计划。如上所述，任务管理器 110 执行新创建子计划中的任务 631-633。

上述参数范围划分方法可用软件实施，以在计算机上执行。例如，所述软件产生一个或多个计算机程序中的过程，所述计算机程序在一个或多个编程或可编程计算机系统（其可具有多种结构，例如分布式、客户/服务器、或网格）上执行，每个计算机系统包括至少一个处理器、至少一个数据存储系统（包括易失和非易失性存储器和/或存储元件）、至少一个输入装置或端口、及至少一个输出装置或端口。所述软件可产生一较大程序的一个或多个模块，例如，其提供与计算图的设计和配置相关的其它服务。

所述软件可以设置在由通用或专用可编程计算机可读的介质（例如 CD-ROM）上，或者可以通过网络传递（以传播信号编码）到执行该软件的计算机。所有功能可以在专用计算机上执行，或者使用诸如协处理器的专用硬件来执行。所述软件可以以分布式来实现，在该分布式中通过不同的计算机来执行由所述软件指定的计算的不同部分。每个这种计算机程序优选地存储或者下载到由通用或专用可编程计算机可读的存储介质或装置（例如，固态存储器或介质、或者磁或光介质）中，用于在由计算机系统读取该存储介质或装置时对计算机进行配置和操作，以执行这里所述的过程。本发明系统还可以认为是作为由计算机程序配置的计算机可读存储介质来实现，其中这样配置的存储介质使得计算机系统以专用或者预定的方式来操作，以执行这里所述的功能。

应该理解以上的描述是示例的目的，而不是对由所附权利要求所限定的本发明范围的限制。其它实施例也落于以下权利要求书的范围中。

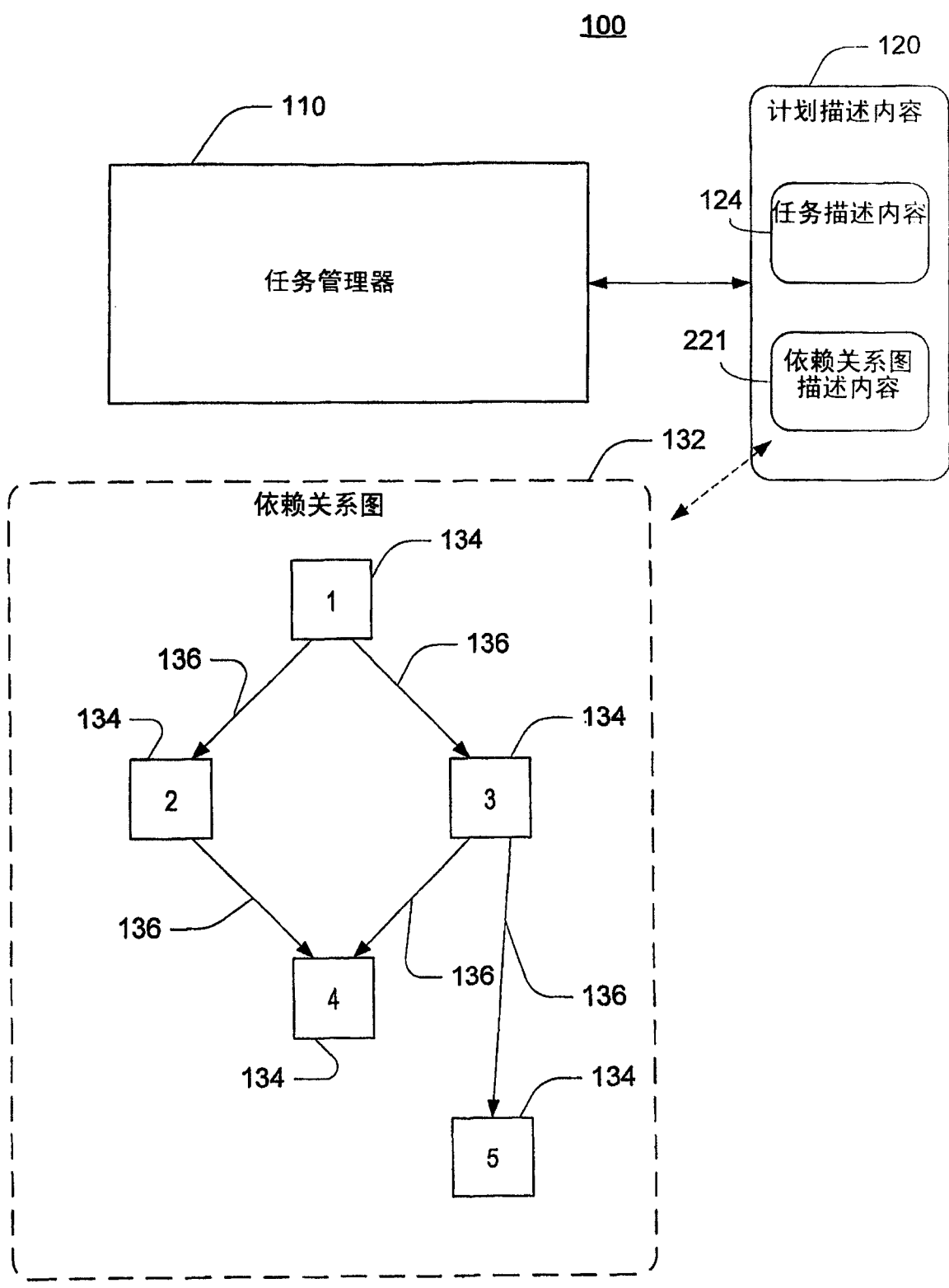


图 1

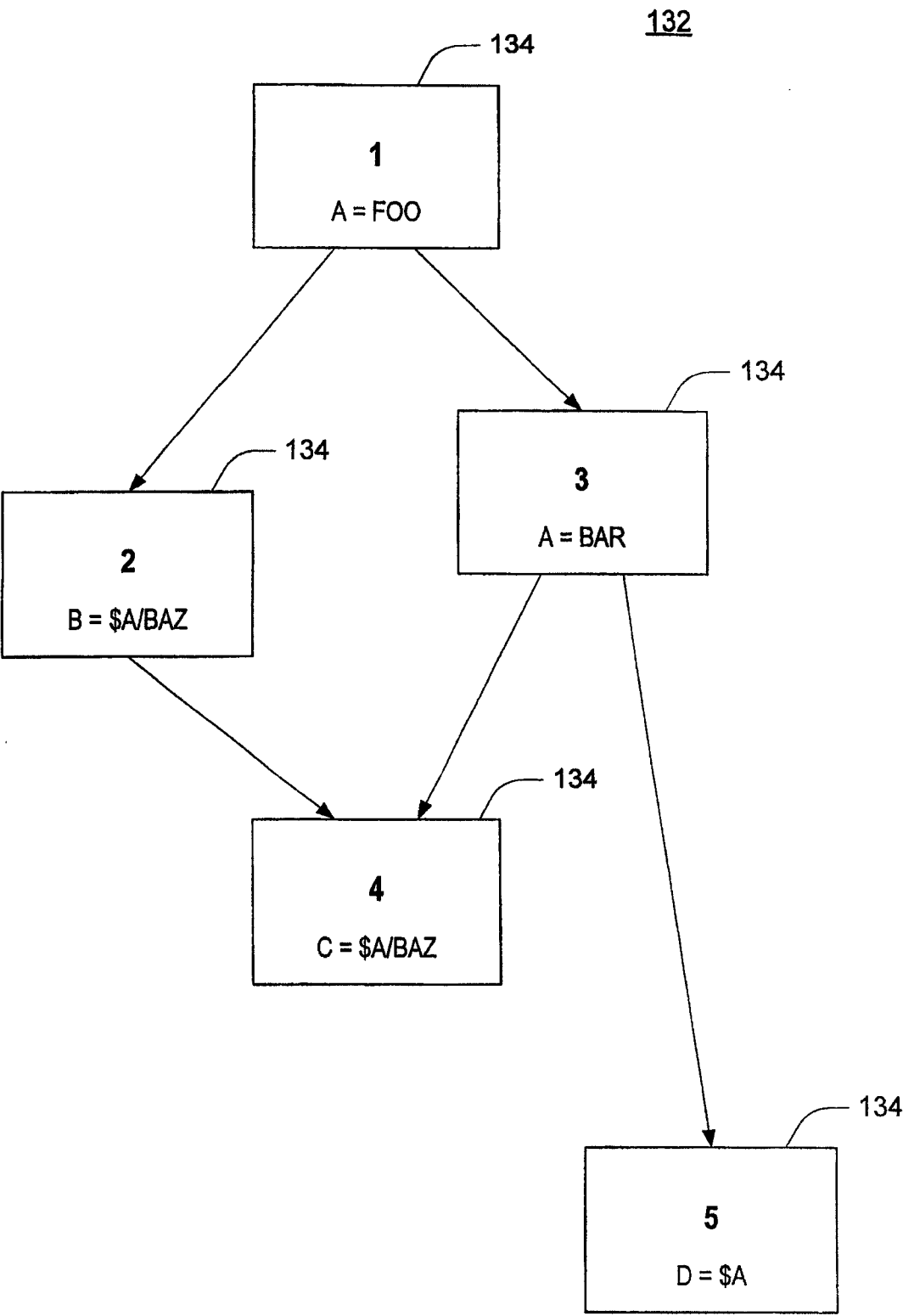


图 2

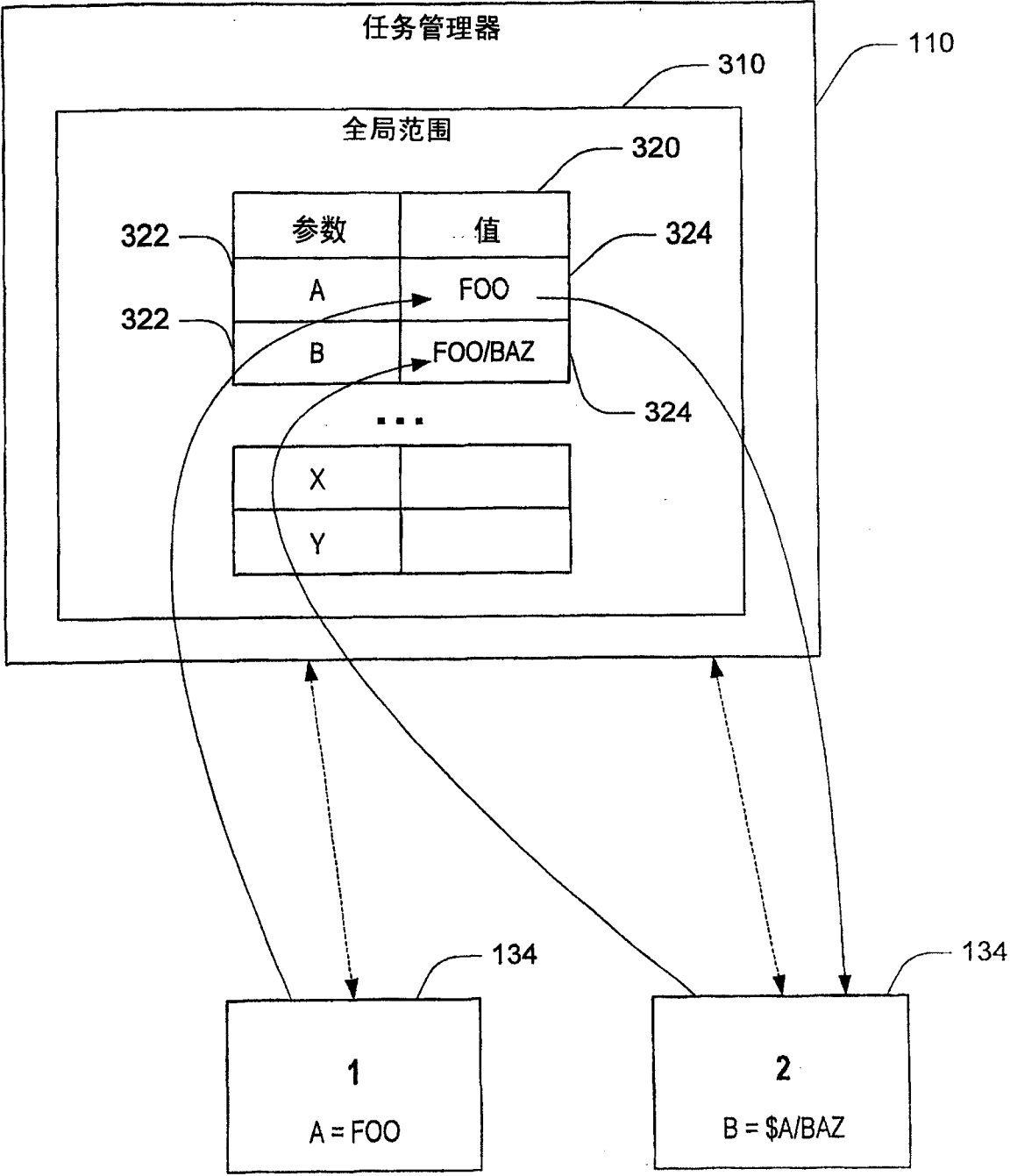


图 3

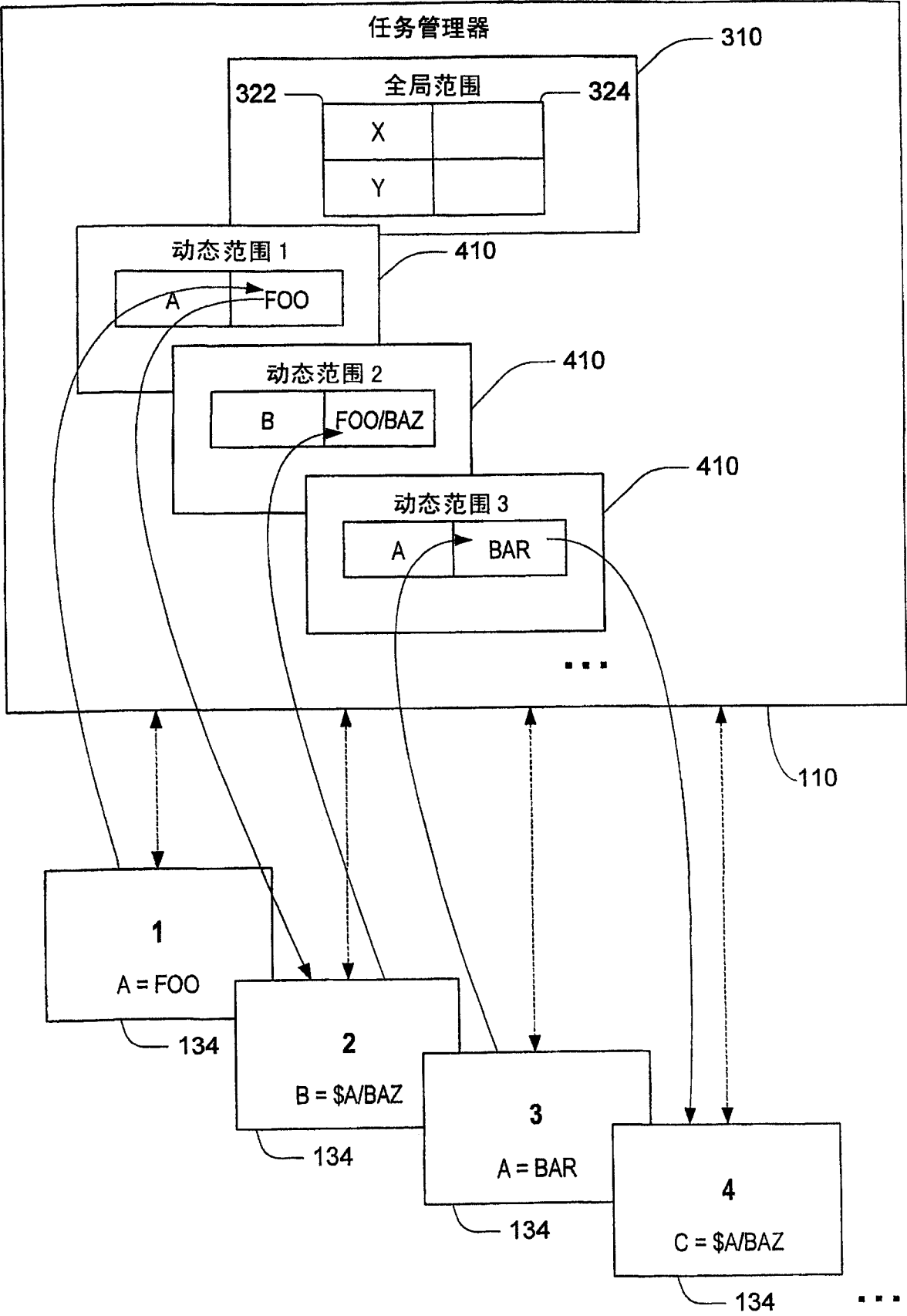


图 4A

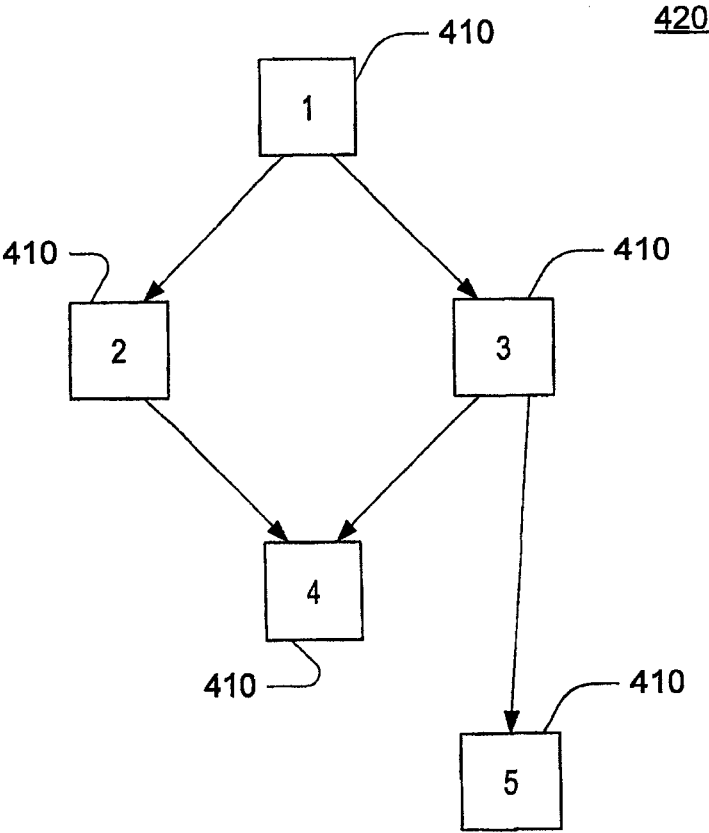


图 4B

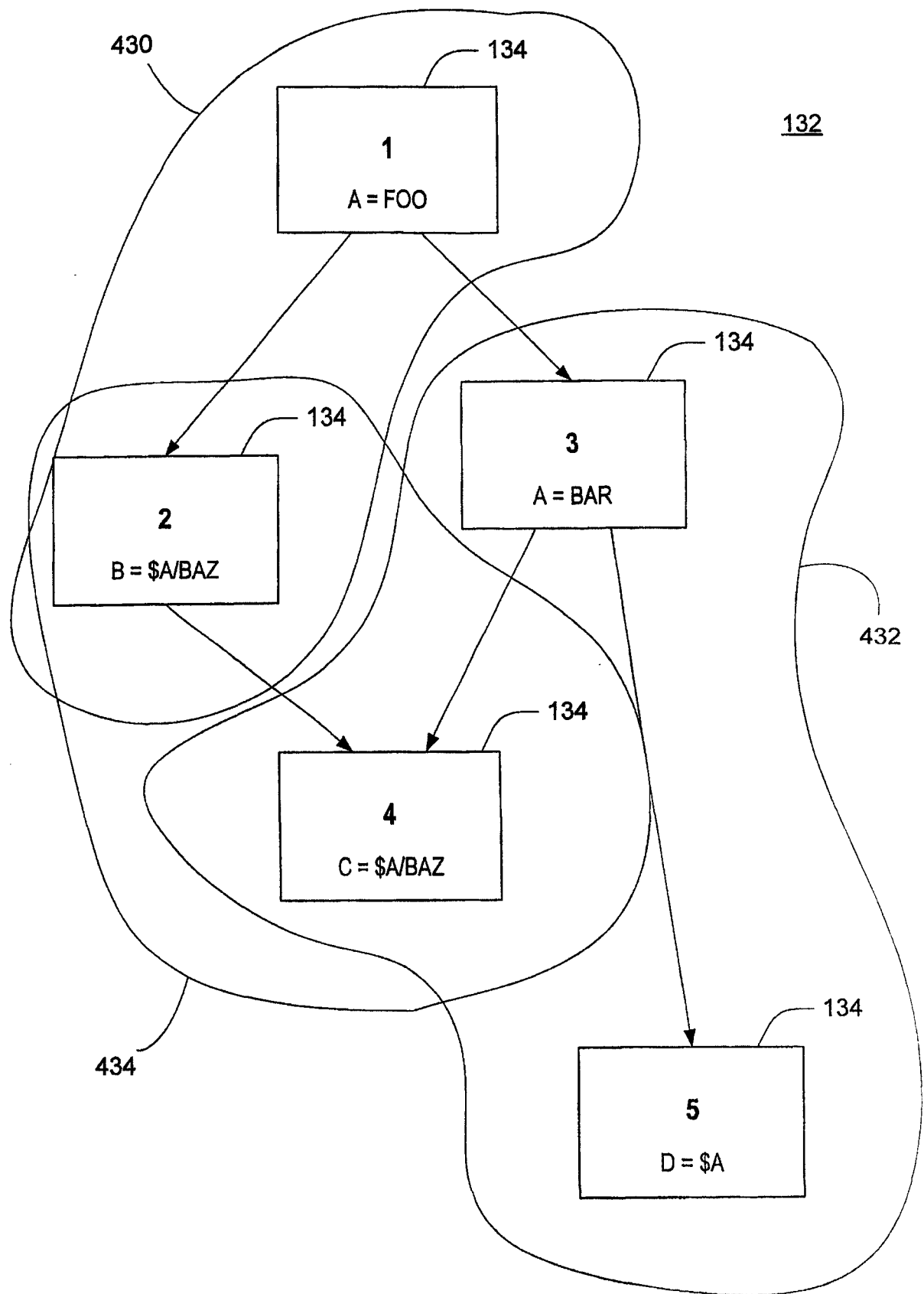


图 4C

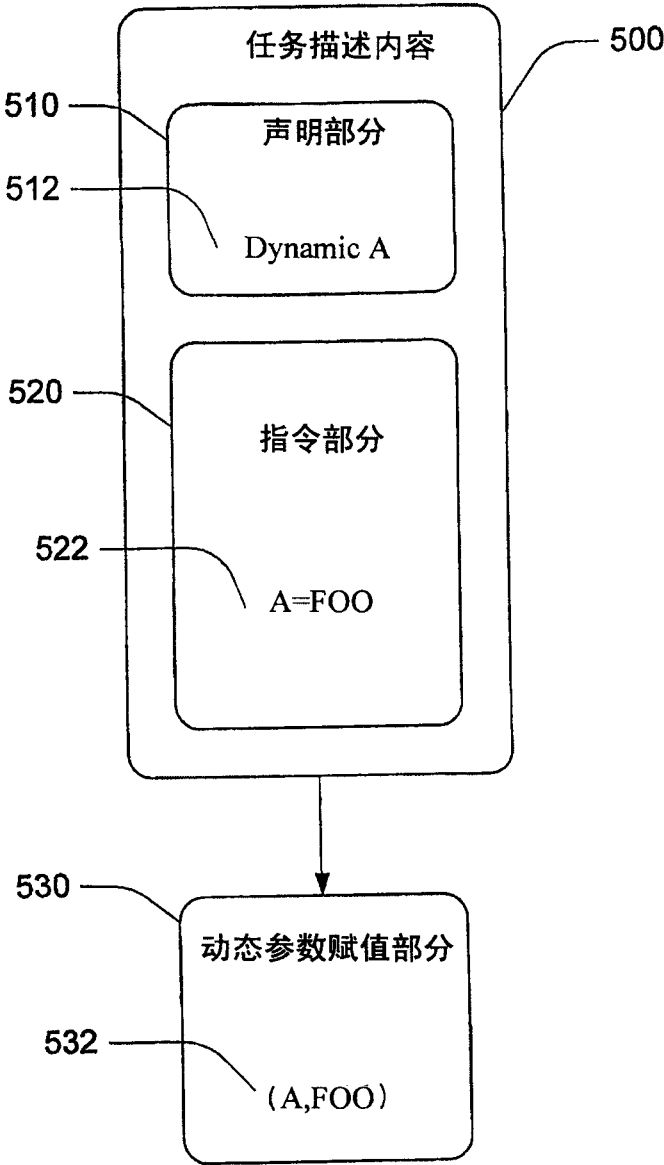


图 5

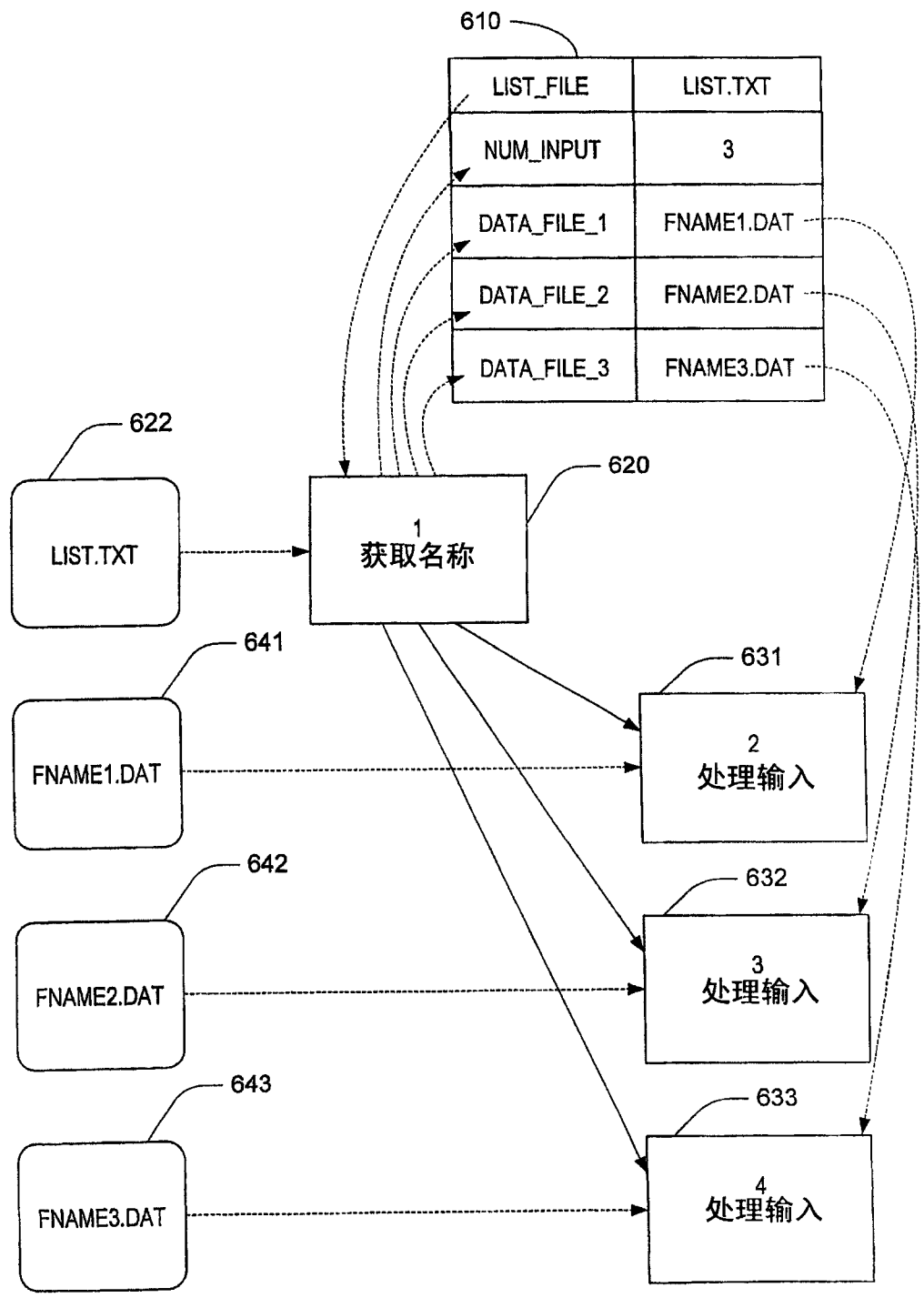


图 6A

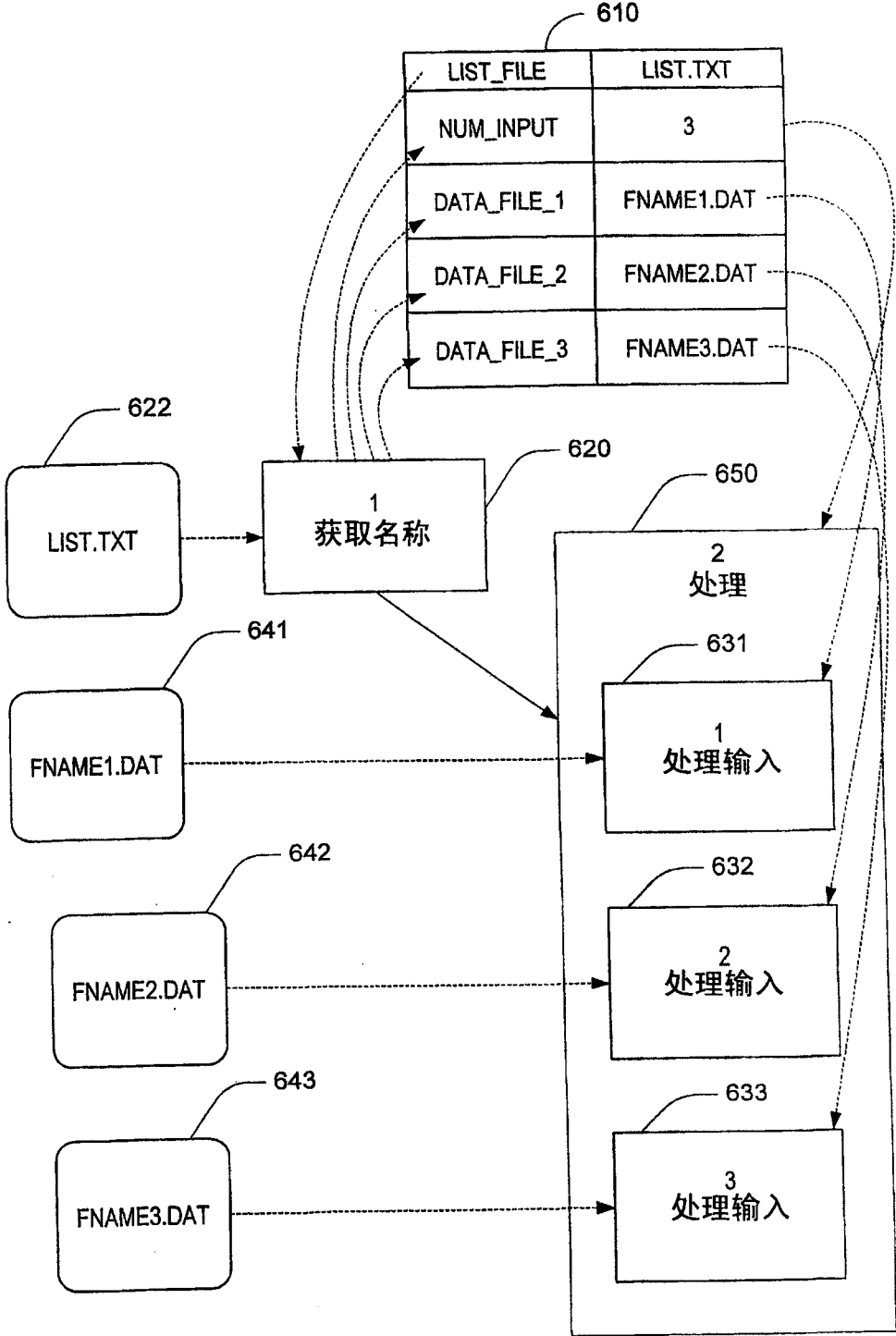


图 6B