



[12] 发明专利申请公开说明书

[21] 申请号 200510123295.2

[43] 公开日 2006年6月7日

[11] 公开号 CN 1783011A

[22] 申请日 2005.11.15

[21] 申请号 200510123295.2

[30] 优先权

[32] 2004.12.2 [33] US [31] 11/002,550

[71] 申请人 国际商业机器公司

地址 美国纽约

[72] 发明人 K·M·奥布赖恩

J·K·P·奥布赖恩

[74] 专利代理机构 北京市中咨律师事务所

代理人 于静 李峥

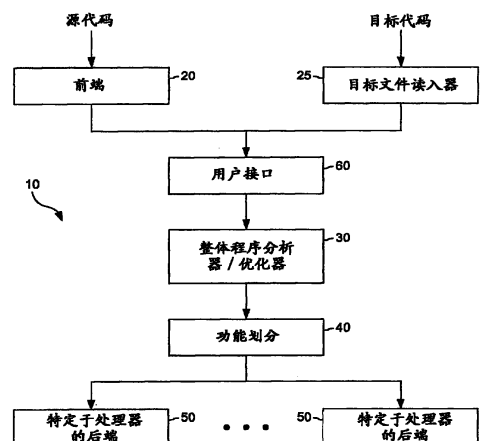
权利要求书 4 页 说明书 9 页 附图 2 页

[54] 发明名称

用于异构型多处理系统的计算机程序功能划分系统和方法

[57] 摘要

本发明提供了一种用于为异构型多处理器系统进行计算机程序功能划分的方法。识别包括一个或多个异类处理节点的计算机系统的至少一个系统参数。接收包括将在该计算机系统上运行的程序的计算机程序代码。根据所接收的计算机程序代码生成整体程序表示。根据该整体程序表示识别至少一个单入口单出口 (SESE) 区域。根据所识别出的 SESE 区域和至少一个系统参数, 识别至少一个特定于节点的 SESE 区域。将每个特定于节点的 SESE 区域组合为特定于节点的子程序。根据指定的节点特性, 编译每个特定于节点的子程序。根据特定于节点的子程序修改计算机程序代码, 并编译该修改后的计算机程序代码。



1. 一种用于为异构型多处理器系统进行计算机程序功能划分的方法，包括：

识别包括一个或多个异类处理节点的计算机系统的至少一个系统参数；

接收包括将在该计算机系统中运行的程序的计算机程序代码；

根据所接收的计算机程序代码生成程序表示；

根据该整体程序表示识别至少一个单入口单出口（SESE）区域；

根据所识别出的 SESE 区域和上述至少一个系统参数，识别至少一个特定于节点的 SESE 区域；

将每个特定于节点的 SESE 区域组合为特定于节点的子程序；

根据这些特定于节点的子程序，修改上述计算机程序代码的非特定于节点的部分；以及

根据指定的节点特性，编译包括每个特定于节点的子程序的修改后的计算机程序代码。

2. 如权利要求 1 所述的方法，还包括：

识别与所识别出的特定于节点的 SESE 区域邻近的 SESE 区域，其中这些识别出的邻近 SESE 区域被配置为针对指定节点而被编译；以及

将至少一个邻近 SESE 区域与至少一个特定于节点的 SESE 区域组合为特定于节点的子程序。

3. 如权利要求 1 所述的方法，其中生成程序表示包括对上述整个程序的过程间分析。

4. 如权利要求 1 所述的方法，其中上述计算机系统包括宽带处理器体系结构系统。

5. 如权利要求 1 所述的方法，还包括生成程序结构树。

6. 如权利要求 1 所述的方法，其中修改上述计算机程序代码包括生成定序指令。

7. 如权利要求 1 所述的方法, 还包括:

接收来自用户的输入; 以及

根据所接收的用户输入, 识别上述计算机系统的至少一个系统参数。

8. 一种用于为异构型多处理器系统进行计算机程序代码功能划分的计算机程序产品, 该计算机程序产品具有其上包括有计算机程序的介质, 该计算机程序包括:

用于识别包括一个或多个异类处理节点的计算机系统的至少一个系统参数的计算机程序代码;

用于接收包括将在该计算机系统中运行的程序的计算机程序代码的计算机程序代码;

用于根据所接收的计算机程序代码生成程序表示的计算机程序代码;

用于根据该整体程序表示识别至少一个单入口单出口 (SESE) 区域的计算机程序代码;

用于根据所识别出的 SESE 区域和上述至少一个系统参数识别至少一个特定于节点的 SESE 区域的计算机程序代码;

用于将每个特定于节点的 SESE 区域组合为特定于节点的子程序的计算机程序代码;

用于根据这些特定于节点的子程序修改上述计算机程序代码的计算机程序代码; 以及

用于根据指定的节点特性编译包括每个特定于节点的子程序的修改后的计算机程序代码的计算机程序代码。

9. 如权利要求 8 所述的计算机程序产品, 还包括:

用于识别与所识别出的特定于节点的 SESE 区域邻近的 SESE 区域的计算机程序代码, 其中这些识别出的邻近 SESE 区域被配置为针对指定节点而被编译; 以及

用于将至少一个邻近 SESE 区域与至少一个特定于节点的 SESE 区域组合为特定于节点的子程序的计算机程序代码。

10. 如权利要求 8 所述的计算机程序产品, 其中生成整体程序表示包

括对上述整个程序的过程间分析。

11. 如权利要求 8 所述的计算机程序产品, 其中上述计算机系统包括宽带处理器体系结构系统。

12. 如权利要求 8 所述的计算机程序产品, 还包括用于生成程序结构树的计算机程序代码。

13. 如权利要求 8 所述的计算机程序产品, 其中修改上述计算机程序代码包括生成定序指令。

14. 如权利要求 8 所述的计算机程序产品, 还包括:

用于从用户接收输入的计算机程序代码; 以及

用于根据所接收的用户输入识别上述计算机系统的至少一个系统参数的计算机程序代码。

15. 一种编译器, 包括:

目标文件读入器模块, 其被配置为识别计算机系统的至少一个系统参数;

整体程序分析器及优化器模块, 其连接到上述目标文件读入器模块, 并被配置为接收包括将在该计算机系统上运行的程序的计算机程序代码, 根据所接收的计算机程序代码生成整体程序表示, 根据该整体程序表示识别至少一个单入口单出口 (SESE) 区域, 并根据所识别出的 SESE 区域和上述至少一个系统参数识别至少一个特定于节点的 SESE 区域; 以及

功能划分模块, 其连接到上述整体程序分析器及优化器模块, 并被配置为将每个特定于节点的 SESE 区域组合为特定于节点的子程序, 根据这些特定于节点的子程序修改上述计算机程序代码, 根据指定的节点特性编译每个特定于节点的子程序, 并编译上述修改后的计算机程序代码。

16. 如权利要求 15 所述的编译器, 其中上述整体程序分析器及优化器模块还被配置为识别与所识别出的特定于节点的 SESE 区域邻近的 SESE 区域, 其中这些识别出的邻近 SESE 区域被配置为针对指定节点而被编译, 并将至少一个邻近 SESE 区域与至少一个特定于节点的 SESE 区域组合为特定于节点的子程序。

17. 如权利要求 15 所述的编译器，其中生成整体程序表示包括过程间分析。

18. 如权利要求 15 所述的编译器，其中上述计算机系统包括宽带处理器体系结构系统。

19. 如权利要求 15 所述的编译器，其中上述整体程序分析器及优化器模块还被配置为生成程序结构树。

20. 如权利要求 15 所述的编译器，其中修改上述计算机程序代码包括生成定序指令。

21. 如权利要求 15 所述的编译器，还包括被配置为接收来自用户的输入的用户接口；以及

其中上述整体程序分析器及优化器模块还被配置为根据所接收的用户输入，识别上述计算机系统的至少一个系统参数。

用于异构型多处理系统的计算机程序 功能划分系统和方法

技术领域

本发明总体涉及计算机程序开发领域，更具体地，涉及一种用于为异构型多处理系统进行计算机程序功能划分的系统和方法。

背景技术

现代的计算机系统经常使用复杂的体系结构，这些体系结构可包括具有不同的配置和能力的多个处理单元。在普通的配置内，所有处理单元都是同样或同类的。比较特殊地，可使用两个或多个不同的或异类的处理单元。在此情况下，不同的处理器将具有专门为特定任务而设计的指令集或能力。每个处理器可更适合不同类型的处理，并且甚至一些处理器固有地不能完全执行特定功能。在此情况下，当需要时，那些功能必须在能够执行它们的处理器上执行，并且最佳地是在最适合于该任务的处理器上执行，如果这样做不会在总体上损害系统的性能的话。

通过在系统的处理单元上执行专门设计的软件（这里称作计算机程序或代码）这一过程，可实现计算机系统的使用。这些代码通常是由程序员以计算机语言编写而生成，并且使用编译器来准备在计算机系统上执行。程序设计任务的简易性以及该代码在计算机系统上的最终执行的效率受到编译器所提供的功能的极大影响。许多现代的简单编译器为单个处理器生成缓慢执行的代码。已构造出了其他的编译器，其为同构型多处理系统内的一个或多个处理器生成可相当快地执行代码。

通常，为了准备程序在异构型多处理系统上执行，典型的现代系统需要程序员使用一些编译器，并费力地组合这些工作的结果以构建最终的代

码。为此，程序员必须按以下方式将源程序分区，即使用适合的处理器执行代码的不同功能。具体地，当该系统内的某些处理器不能执行特定功能时，被分区的程序必须在提供此能力的处理器上执行那些功能。此外，即使当设计了多种处理器类型来执行相同功能，以使系统性能最佳时，划分也应该将那些功能指派给可最迅速地执行所述功能的处理器。

因此，需要一种用于为异构型多处理系统进行计算机程序功能划分的系统和/或方法，其可解决至少一些与传统系统和方法相关联的问题和缺陷。

发明内容

本发明提供了一种用于为异构型多处理系统进行计算机程序功能划分的方法。识别包括一个或多个完全不同的处理节点的计算机系统的至少一个系统参数。接收包括将在该计算机系统上运行的程序的计算机程序代码。根据所接收的计算机程序代码生成整体程序表示。根据该整体程序表示识别至少一个单入口单出口（SESE）区域。根据所识别出的 SESE 区域和至少一个系统参数，识别至少一个特定于节点的 SESE 区域。将每个特定于节点的 SESE 区域组合为特定于节点的子程序。根据指定的节点特性，编译每个特定于节点的子程序。根据特定于节点的子程序修改计算机程序代码，并编译该修改后的计算机程序代码。

附图说明

参照下文结合附图进行的说明，可更充分地理解本发明及其优点，在附图中：

图 1 是示出计算机程序功能划分系统的框图；以及

图 2 是示出计算机程序功能划分方法的流程图。

具体实施方式

在下面的讨论中，说明了众多特定的细节以便提供对本发明的彻底理

解。但是，本领域内的技术人员应理解，本发明可在不具有这种特定细节的情况下实现。在其他情况下，已用示意图或框图形式示出了公知的组件，以便不会以不必要的细节来混淆本发明。另外，在很大程度上已省略了关于网络通信、电磁信号传输技术、用户接口或输入/输出技术等细节，这是因为这些细节不被认为对于充分理解本发明来说是必须的，而是被认为处于本领域普通技术人员的理解范围内。

还应指出，除非另外说明，否则文中所述的所有功能可以硬件或软件来实现，或者以它们的某些组合来实现。但是，在优选实施例内，除非另外说明，否则这些功能是由诸如计算机或电子数据处理器的处理器根据诸如计算机程序代码、软件的代码、和/或由被编码为执行这些功能的集成电路来实现的。

参照图 1，参考标号 10 总体指编译器。编译器 10 是电路或其他合适的逻辑，并且其被配置为计算机程序代码编译器。在一个特定实施例内，如下面更详细地说明的，编译器 10 是被配置成将源代码编译成目标代码的软件程序。通常，编译器 10 被配置成通过用户接口 60 在前端代码模块 20 内接收源代码，并通过目标文件读入器 25 接收目标代码。然后，使用整体程序分析器及优化器 30 以及功能划分模块 40 提供的参数来处理接收到的代码，以生成特定于处理器的后端代码 50，这在下文将更详细地说明。

具体地，在所示实施例内，编译器 10 包括用户接口 60。用户接口 60 是电路或其他适合的逻辑，并且其被配置成接收来自用户的输入。在一个实施例内，用户接口 60 是在程序源代码内插入的“编译指示 (pragmas)”命令或指示与命令行上或者“编译文件 (makefile)”命令或脚本内提供的编译器选项标志的组合。在另一实施例内，可使用图形用户接口来提供此信息。本领域的技术人员应理解，还可使用其他配置。

编译器 10 包括前端代码模块 20。前端代码模块 20 是电路或其他适合的逻辑，并且其被配置成读取源代码并识别编译后的代码将在其上执行的计算机系统的特定参数。编译器 10 还包括目标文件读入器模块 25。目标文件读入器模块 25 是电路或其他适合的逻辑，且其被配置成读取目标代码

并识别编译后的代码将在其上执行的计算机系统的特定参数。通常，目标代码是编译器 10 先前处理前端代码模块 20 接收到的源代码，并存储关于通过该编译器内的分析得出的所述源代码的信息的保存结果。在一个特定实施例内，目标文件读入器模块 25 是软件程序，且其被配置成识别并映射编译后的代码将在其上执行的计算机系统（即“目标”系统）的各个处理节点。另外，目标文件读入器模块 25 还被配置成识别已识别出的节点的处理能力。

编译器 10 还包括整体程序分析器及优化器模块 30。整体程序分析器及优化器模块 30 是电路或其他适合的逻辑，且其被配置成分析接收到的源和/或目标代码，这在下文将更详细地说明。在一个特定实施例内，整体程序分析器及优化器模块 30 是软件程序且其被配置成创建接收到的源和/或目标代码的整体程序表示并识别计算机程序代码的特定于节点的段。因此，通常，整体程序分析器及优化器模块 30 可被配置成分析整个计算机程序源代码，即接收的源和/或目标代码，以识别源和/或目标代码的需要在特定类型的处理节点上处理的段，并将识别出的段隔离成子程序，这些子程序随后可针对特定需要的处理节点（即“目标”节点）而被编译。如文中使用的，本领域内的技术人员应理解，整个计算机程序源代码是构成不连续的计算机程序的一组计算机程序代码行。

具体地，在一个实施例内，整体程序分析器及优化器模块 30 被配置成接收源和/或目标代码，并创建接收到的代码的整体程序表示。如这里所用的，整体程序表示是构成整个计算机程序源代码的各个代码段的表示。在一个实施例内，整体程序分析器及优化器模块 30 被配置成对接收到的源和/或目标代码执行过程间分析，以创建整体程序表示。本领域内的技术人员应理解，还可使用其他方法来创建该接收到的计算机程序源代码的整体程序表示。

在一个实施例内，整体程序分析器及优化器模块 30 还被配置成识别该整体程序表示内的单入口单出口（SESE）区域。通常，本领域内的技术人员应理解，SESE 区域是计算机程序内对于该计算机程序的流程总体上遵

守某些公知特性的段。因此，可使用识别出的 SESE 区域来帮助将程序划分成子程序。本领域内的技术人员应理解，通过合适的变换可将没有严格定义的 SESE 区域的其他适合的程序段转换为 SESE 区域。在一个特定实施例内，整体程序分析器及优化器模块 30 被配置成通过构造和分析程序结构树来识别 SESE 区域。本领域内的技术人员应理解，还可使用其他方法来识别整体程序表示内的 SESE 区域。

另外，整体程序分析器及优化器模块 30 可被配置成识别包含需要在特定类型的处理节点上处理的功能的 SESE 区域。具体地，整体程序分析器及优化器模块 30 可被配置成从目标文件读入器 25 和/或用户接口 60 接收系统参数和/或系统参数信息，以识别需要在特定类型的处理节点上处理的 SESE 区域，并使识别出的 SESE 区域与这些系统参数所描述的特定计算机系统节点的能力相关联。因此，通常，在一个实施例内，整体程序分析器及优化器模块 30 被配置成根据目标系统（更具体地，目标节点）的系统参数，识别特定于节点的 SESE 区域。

另外，整体程序分析器及优化器模块 30 可被配置成识别在整个流程内与识别出的特定于节点的 SESE 区域邻近的 SESE 区域。因此，整体程序分析器及优化器模块 30 还可被配置成识别可根据目标计算机系统的系统效率而与识别出的特定于节点的 SESE 区域组合在一起的邻近 SESE 区域。因此，整体程序分析器及优化器模块 30 可被配置成通过将识别出的邻近 SESE 区域与识别出的特定于节点的 SESE 区域组合在一起，而提供增加的过程效率。本领域内的技术人员应理解，可根据特定处理节点处理识别出的邻近 SESE 区域的能力，来使识别出的邻近 SESE 区域与识别出的特定于节点的 SESE 区域组合在一起。

编译器 10 还包括功能划分模块 40。功能划分模块 40 是电路或其他适合的逻辑，且其总体上被配置成将识别出的特定于节点的 SESE 区域划分成子程序，并针对特定子程序将在其上执行的目标节点来编译该子程序。因此，在一个特定实施例内，功能划分模块 40 被配置成将识别出的特定于节点的 SESE 区域组合为分立的子程序。在一个实施例内，功能划分模块

40 可配置成将每个识别出的特定于节点的 SESE 区域组合为一个单独的子程序。在可选实施例内，功能划分模块 40 还被配置成将一个或多个识别出的特定于节点的 SESE 区域组合成一个组合子程序。本领域内的技术人员应理解，将一个或多个识别出的特定于节点的 SESE 区域组合成一个组合子程序是以程序设计对于组合所述区域的适合性为条件的。例如，本领域内的技术人员应理解，当一个或多个识别出的特定于节点的 SESE 区域位于整个流程内的需要不兼容的中间过程步骤的位置时，这些区域不适于被组合成一个组合子程序。

另外，功能划分模块 40 还可被配置成根据通过用户接口 60 从用户接收到的输入，对识别出的特定于节点的 SESE 区域和/或识别出的邻近 SESE 区域进行组合。因此，例如，在一个实施例内，编译器 10 可被配置成通过用户接口 60 将识别出的特定于节点的 SESE 区域和/或识别出的邻近 SESE 区域呈现给用户，并根据用户输入对所呈现的区域进行组合。

在一个实施例内，功能划分模块 40 还被配置成根据组合子程序修改接收到的源和/或目标代码。在一个特定实施例内，功能划分模块 40 被配置成通过定序指令来修改接收到的源和/或目标代码。具体地，功能划分模块 40 可被配置成在接收到的源和/或目标代码内插入定时占位，藉此编排子程序的执行定时。例如，插入定时占位可包括预加载特定的代码段以帮助确保正确的程序功能，帮助使代码执行中的延时最小化或消除，以及出于其他适合的定时考虑。本领域内的技术人员应理解，还可使用其他配置。

功能划分模块 40 还被配置成根据编译后的特定于处理器的后端代码段将在其上执行的特定处理节点（即目标节点），将接收到的源和/或目标代码编译成一个或多个特定于处理器的后端代码段 50。因此，针对支持特定于处理器的后端代码段 50 内包含的特定功能所需的特定于节点的功能，来编译这些代码段。

因此，通常，在操作时，编译器 10 接收计算机程序代码（源和/或目标代码），并根据目标系统参数分析接收到的计算机程序代码的特定于节点的 SESE 区域。将特定于节点的 SESE 区域组合为子程序，并针对它们

的目标节点以及适当的邻近 SESE 区域（如果有的话）编译这些 SESE 区域。因此，编译器 10 可配置成使某些时间密集的程序设计活动（诸如识别特定于节点的 SESE 区域并将识别出的特定于节点的 SESE 区域组合成子程序）自动进行，从而除去程序员的负担，否则程序员将不得不执行这些任务。因此，编译器 10 可配置成将计算机程序代码划分为特定于节点的子程序，针对这些子程序将在其上执行的特定类型的目标节点来编译它们，对指令定序以维护过程定时和控制。

参照图 2，参考标号 200 总体指示示出计算机程序分区方法的流程图。该过程在步骤 205 开始，其中识别系统参数。此步骤可由例如图 1 的目标文件读入器模块 25 执行和/或通过图 1 的用户接口 60 接收到的用户输入来执行。在下一步骤 210，接收或扫描输入将被分析的计算机程序代码。此步骤可由例如图 1 的编译器前端和/或前端代码模块 20 或目标文件读入器模块 25 执行。本领域内的技术人员应理解，接收或扫描输入将被分析的代码包括检索硬盘驱动器或其他适合的存储设备上存储的数据，并将这些数据加载到系统存储器中。另外，在编译器前端和/或前端代码模块 20 的情况下，此步骤还可包括对源语言程序进行分析，并产生中间形式代码。在目标文件读入器模块 25 的情况下，此步骤可包括从计算机程序代码的目标代码文件提取中间表示。

在下一步骤 215，根据接收到的计算机程序代码生成整体程序表示。此步骤可由例如图 1 的整体程序分析器及优化器模块 30 执行。本领域内的技术人员应理解，此步骤可包括进行过程间分析。在下一步骤 220，根据该整体程序表示识别 SESE 区域。此步骤可由例如图 1 的整体程序分析器及优化器模块 30 执行。本领域内的技术人员应理解，此步骤可包括生成程序结构树。

在下一步骤 225，根据识别出的系统参数和识别出的 SESE 区域，识别特定于节点的 SESE 区域。此步骤可由例如图 1 的整体程序分析器及优化器模块 30 执行。如上所述，此步骤还可包括识别适于与识别出的特定于节点的 SESE 区域组合在一起的邻近 SESE 区域。在下一步骤 230，将识

别出的特定于节点的 SESE 区域组合为一个或多个特定于节点的子程序或分区。此步骤可由例如图 1 的功能划分模块 40 执行。如上所述,此步骤可包括将每个特定于节点的 SESE 区域组合为其自身的子程序或分区,和/或将一个或多个特定于节点的 SESE 区域组合成组合子程序或分区。

在下一步骤 235,修改接收到的计算机程序代码,以包括对于一个或多个子程序的定序指令。此步骤可由例如图 1 的功能划分模块 40 执行。本领域内的技术人员应理解,此步骤可包括根据需要在处理器之间传送代码和/或数据的指令,和检查在其他处理器上执行的分区的完成并执行其他适当的动作的指令。

在下一步骤 240,为每个分区生成特定于处理器的代码,并且该过程结束。此步骤可由例如适于特定节点类型的图 1 的特定于处理器的后端模块 50 执行。因此,可将计算机程序划分为子程序,这些子程序被标记以便针对特定节点类型进行编译,对修改结果定序以编排目标系统内的各个节点类型之间的通信。因此,可将为具有异类或异构型处理组件的多处理器系统设计的计算机程序代码配置成考虑需要在特定类型的节点上执行的特定功能。

因此,本发明提供了在具有至少两个异类处理器的异构型多处理器系统的节点之间进行计算机程序功能划分。将编译器配置成生成特定于每个异类处理器的代码。通常,编译器读取可分为多个文件并且可在不同的时间处理的源程序,并构建整个程序的模型或表示。然后,根据多个异类处理器或节点的特性分析此模型。在该模型内识别特定于或更适于在特定处理器类型上执行的一个或多个功能。以优化程序的总执行时间的方式,识别完全包含识别出的功能的单入口单出口区域(SESE),该区域被提取到特定于节点的子程序中。用适当的链接或定序代码代替被提取出的 SESE 区域,以确保正确的和有效的程序执行。针对所得到的每个特定于处理器的分区或子程序的各自的处理器编译它们,并合并所得的编译结果以形成最终程序,以在异构型多处理器系统上执行。

上面公开的特定实施例仅是说明性的,因为本发明可按不同但等效的

方式来修改和实现，这些方式对于受益于本文的教导的本领域技术人员来说是显而易见的。此外，除了下面权利要求所描述的之外，本发明并不局限于这里所示的结构或设计细节。因此很明显，上面公开的特定实施例可被替换或修改，所有这样的变型均被认为落入本发明的范围和精神之内。因此，下面的权利要求中说明了所寻求的保护范围。

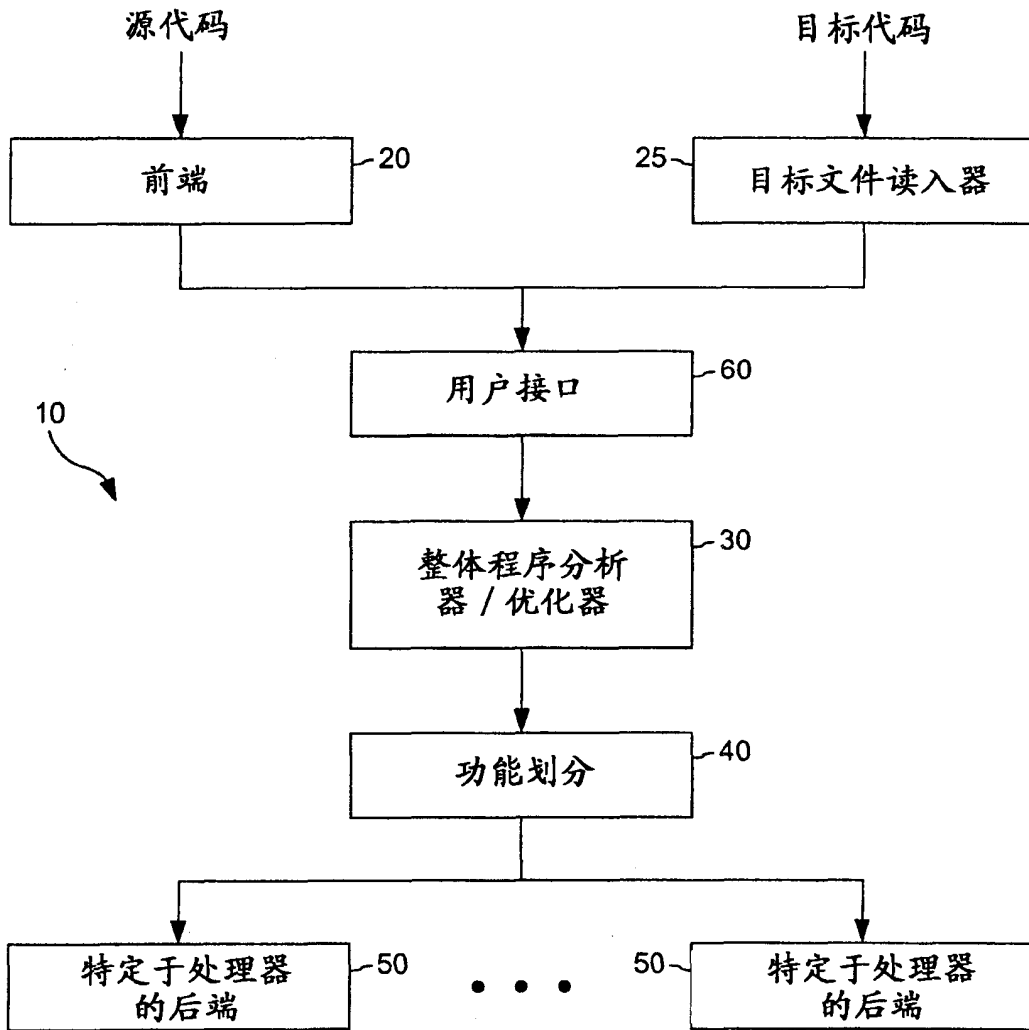


图 1

图 2

