



(12) 发明专利

(10) 授权公告号 CN 119883507 B

(45) 授权公告日 2025.08.22

(21) 申请号 202411943710.8

(22) 申请日 2024.12.26

(65) 同一申请的已公布的文献号

申请公布号 CN 119883507 A

(43) 申请公布日 2025.04.25

(73) 专利权人 中煤科工开采研究院有限公司

地址 101399 北京市顺义区中关村科技园
区顺义园临空二路1号

(72) 发明人 吕依濛

(74) 专利代理机构 北京清亦华知识产权代理事

务所(普通合伙) 11201

专利代理师 白雪静

(51) Int. Cl.

G06F 9/455 (2018.01)

G06F 9/50 (2006.01)

(56) 对比文件

CN 115543547 A, 2022.12.30

CN 101963918 A, 2011.02.02

审查员 李爽

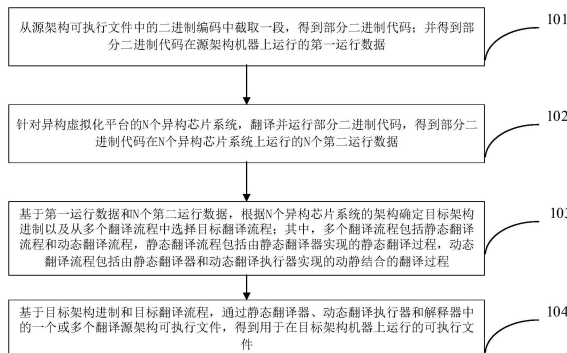
权利要求书3页 说明书10页 附图3页

(54) 发明名称

异构虚拟化平台虚拟机迁移方法及系统

(57) 摘要

本申请提出一种异构虚拟化平台虚拟机迁移方法及系统,其中,方法包括:从源架构可执行文件中的二进制编码中截取一段,得到部分二进制代码;并得到部分二进制代码在源架构机器上运行的第一运行数据;针对异构虚拟化平台的N个异构芯片系统,翻译并运行部分二进制代码,得到部分二进制代码在N个异构芯片系统上运行的N个第二运行数据;基于第一运行数据和N个第二运行数据,根据N个异构芯片系统的架构确定目标架构进制和目标翻译流程;基于目标架构进制和目标翻译流程,通过静态翻译器、动态翻译执行器和解释器中的一个或多个翻译源架构可执行文件,得到用于在目标架构机器上运行的可执行文件;从而提高异构虚拟化平台虚拟机迁移效果。



1. 一种异构虚拟化平台虚拟机迁移方法,其特征在于,所述方法包括以下步骤:

从源架构可执行文件中的二进制编码中截取一段,得到部分二进制代码;并得到所述部分二进制代码在源架构机器上运行的第一运行数据;

针对所述异构虚拟化平台的N个异构芯片系统,翻译并运行所述部分二进制代码,得到所述部分二进制代码在所述N个异构芯片系统上运行的N个第二运行数据;

基于所述第一运行数据和所述N个第二运行数据,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;其中,所述多个翻译流程包括静态翻译流程和动态翻译流程,所述静态翻译流程包括由静态翻译器实现的静态翻译过程,所述动态翻译流程包括由静态翻译器和动态翻译执行器实现的动静结合的翻译过程;所述第一运行数据包括所述部分二进制代码在源架构机器上运行的第一执行时间,所述第二运行数据包括所述部分二进制代码在第n个异构芯片系统上运行的第二执行时间、CPU资源占用率、内存资源占用率和存储资源占用值;其中, $1 \leq n \leq N$;所述基于所述第一运行数据和所述N个第二运行数据,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;包括:基于所述第一执行时间和所述部分二进制代码在第n个异构芯片系统上运行的第二执行时间,得到所述部分二进制代码在第n个异构芯片系统上运行的执行效率;基于所述部分二进制代码在第n个异构芯片系统上运行的执行效率、CPU资源占用率、内存资源占用率和存储资源占用值,得到N个异构芯片系统的资源值;基于所述N个异构芯片系统的资源值,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;

基于所述目标架构进制和所述目标翻译流程,通过所述静态翻译器、所述动态翻译执行器和解释器中的一个或多个翻译所述源架构可执行文件,得到用于在目标架构机器上运行的可执行文件。

2. 根据权利要求1所述的方法,其特征在于,所述基于所述第n个异构芯片系统的资源值,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;包括:

基于所述N个异构芯片系统的资源值,确定资源值最大的目标异构芯片系统;

根据所述目标异构芯片系统的架构,确定目标架构进制;

当所述目标异构芯片系统的资源值大于资源值上限阈值时,选择静态翻译流程作为目标翻译流程;

当所述目标异构芯片系统的资源值小于等于资源值上限阈值且大于等于资源值下限阈值时,选择动态翻译流程作为目标翻译流程。

3. 根据权利要求1所述的方法,其特征在于,所述基于所述部分二进制代码在第n个异构芯片系统上运行的执行效率、CPU资源占用率、内存资源占用率和存储资源占用值,得到N个异构芯片系统的资源值;包括:通过第一公式,基于所述执行效率、所述CPU资源占用率、所述内存资源占用率和所述存储资源占用值,得到N个异构芯片系统的资源值,所述第一公式表示如下:

$$V_n = \frac{E_n - \frac{\lambda_1 * C_n + \lambda_2 * M_n + \lambda_3 * S_n}{\lambda_1 + \lambda_2 + \lambda_3}}{\frac{\lambda_1 * C_n + \lambda_2 * M_n + \lambda_3 * S_n}{\lambda_1 + \lambda_2 + \lambda_3}} \quad (1)$$

其中, V_n 为第 n 个异构芯片系统的资源值, λ_1 、 λ_2 、 λ_3 均为调节参数, 根据虚拟化平台的 CPU、内存、存储的实际情况及重要程度进行调节; E_n 为在第 n 个异构芯片系统上运行的执行效率, C_n 为在第 n 个异构芯片系统上运行的 CPU 资源占用率, M_n 为在第 n 个异构芯片系统上运行的内存资源占用率, S_n 为在第 n 个异构芯片系统上运行的存储资源占用值。

4. 根据权利要求 1 所述的方法, 其特征在于, 所述翻译流程还包括解释器翻译流程, 所述解释器翻译流程包括解释器直接运行过程; 所述基于所述目标架构进制和所述目标翻译流程, 通过所述静态翻译器、所述动态翻译执行器和解释器中的一个或多个翻译所述源架构可执行文件, 得到用于在目标架构机器上运行的可执行文件; 包括:

当所述目标翻译流程为静态翻译流程时, 通过所述静态翻译器翻译所述源架构可执行文件, 得到翻译后的目标可执行文件;

当所述目标翻译流程为动态翻译流程时, 通过所述静态翻译器和所述动态翻译执行器以动静结合方式翻译所述源架构可执行文件, 得到翻译后的目标可执行文件;

获取所述静态翻译器的翻译效率;

基于所述翻译效率和预设阈值, 确定新的所述目标架构进制、新的所述目标翻译流程以及是否将所述翻译后的目标可执行文件作为所述用于在目标架构机器上运行的可执行文件中的一个或者多个。

5. 根据权利要求 4 所述的方法, 其特征在于, 所述预设阈值包括第一效率阈值和第二效率阈值, 所述基于所述翻译效率和预设阈值, 确定新的所述目标架构进制、新的所述目标翻译流程以及是否将所述翻译后的目标可执行文件作为所述用于在目标架构机器上运行的可执行文件中的一个或者多个; 包括:

当所述翻译效率小于等于第一效率阈值时, 通过所述解释器直接运行所述源架构可执行文件, 得到用于在目标架构机器上运行的可执行文件;

当所述翻译效率大于所述第一效率阈值且小于所述第二效率阈值、同时所述目标翻译流程为动态翻译流程时, 则通过所述静态翻译器重新翻译所述源架构可执行文件, 得到翻译后的目标可执行文件并返回重新获取所述静态翻译器的翻译效率;

当所述翻译效率大于所述第一效率阈值且小于所述第二效率阈值、同时所述目标翻译流程为静态翻译流程时, 基于所述第一运行数据和第二运行数据, 返回重新从所述 N 个异构芯片系统中确定新的目标架构进制以及目标翻译流程;

当所述翻译效率大于所述第二效率阈值时, 将所述翻译后的目标可执行文件作为所述用于在目标架构机器上运行的可执行文件。

6. 根据权利要求 1 所述的方法, 其特征在于, 所述针对所述异构虚拟化平台的 N 个异构芯片系统, 翻译并运行所述部分二进制代码, 得到所述部分二进制代码在所述 N 个异构芯片系统上运行的 N 个第二运行数据; 包括:

针对所述异构虚拟化平台的 N 个异构芯片系统, 通过所述静态翻译器翻译所述部分二

进制代码,得到分别对应所述N个异构芯片系统的N个可执行代码;

令所述N个可执行代码分别在所述N个异构芯片系统上运行,得到所述部分二进制代码在所述N个异构芯片系统上运行的N个第二运行数据。

7. 根据权利要求4所述的方法,其特征在于,所述通过所述静态翻译器和所述动态翻译执行器以动静结合方式翻译所述源架构可执行文件,得到翻译后的目标可执行文件;包括:

通过所述静态翻译器对所述源架构可执行文件中的二进制编码进行离线翻译和深度优化,得到静态翻译结果;并将所述静态翻译结果传递至所述动态翻译执行器;

通过所述动态翻译执行器装入所述源架构可执行文件中的二进制编码和所述静态翻译结果;对所述源架构可执行文件中的二进制编码中未被翻译的部分代码进行在线翻译和优化,并将在线翻译和优化的结果合并至所述静态翻译结果,得到翻译后的目标可执行文件。

8. 一种异构虚拟化平台虚拟机迁移系统,其特征在于,所述系统包括指令选择器、静态翻译器、动态翻译执行器和解释器,所述指令选择器,用于:

从源架构可执行文件中的二进制编码中截取一段,得到部分二进制代码;并得到所述部分二进制代码在源架构机器上运行的第一运行数据;

针对所述异构虚拟化平台的N个异构芯片系统,翻译并运行所述部分二进制代码,得到所述部分二进制代码在所述N个异构芯片系统上运行的N个第二运行数据;

基于所述第一运行数据和所述N个第二运行数据,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;其中,所述多个翻译流程包括静态翻译流程和动态翻译流程,所述静态翻译流程包括由静态翻译器实现的静态翻译过程,所述动态翻译流程包括由静态翻译器和动态翻译执行器实现的动静结合的翻译过程;所述第一运行数据包括所述部分二进制代码在源架构机器上运行的第一执行时间,所述第二运行数据包括所述部分二进制代码在第n个异构芯片系统上运行的第二执行时间、CPU资源占用率、内存资源占用率和存储资源占用值;其中, $1 \leq n \leq N$;所述基于所述第一运行数据和所述N个第二运行数据,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;包括:基于所述第一执行时间和所述部分二进制代码在第n个异构芯片系统上运行的第二执行时间,得到所述部分二进制代码在第n个异构芯片系统上运行的执行效率;基于所述部分二进制代码在第n个异构芯片系统上运行的执行效率、CPU资源占用率、内存资源占用率和存储资源占用值,得到N个异构芯片系统的资源值;基于所述N个异构芯片系统的资源值,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;

基于所述目标架构进制和所述目标翻译流程,通过所述静态翻译器、所述动态翻译执行器和解释器中的一个或多个翻译所述源架构可执行文件,得到用于在目标架构机器上运行的可执行文件。

9. 一种计算机程序产品,其特征在于,包括计算机程序,该计算机程序被处理器执行时实现权利要求1-7中任一项所述的方法。

异构虚拟化平台虚拟机迁移方法及系统

技术领域

[0001] 本申请涉及异构芯片虚拟机迁移技术领域,尤其涉及一种异构虚拟化平台虚拟机迁移方法及系统。

背景技术

[0002] 中煤科工开采研究院在数字化转型期间建设了大量数字化系统,为了总体管理这些系统,建设了相应的机房和硬件设备,在此基础上采用虚拟化超融合技术为这些系统分配相应的虚拟机资源。

[0003] 之前异构虚拟化平台以X86架构指令集和ARM架构指令集为主,为了响应国家信创产业政策,选择国产CPU芯片如海光、鲲鹏、兆芯、龙芯、申威等,包括指令集架构有X86、ARM、LoognArch、MIPS、SW64、RISC-V等。虚拟机在异构虚拟化平台的不同芯片架构之间进行迁移,以实现较佳的迁移后的虚拟机的运行效果尤为重要,例如从X86到其他国产指令集架构的自动化迁移的技术尤为重要。

发明内容

[0004] 本申请旨在至少在一定程度上解决相关技术中的技术问题之一。

[0005] 为此,本申请的第一个目的在于提出一种异构虚拟化平台虚拟机迁移方法,从而提高异构虚拟化平台虚拟机迁移效果。

[0006] 本申请的第二个目的在于提出一种异构虚拟化平台虚拟机迁移系统。

[0007] 本申请的第三个目的在于提出一种电子设备。

[0008] 本申请的第四个目的在于提出一种计算机可读存储介质。

[0009] 本申请的第五个目的在于提出一种计算机程序产品。

[0010] 为达上述目的,本申请第一方面实施例提出了一种异构虚拟化平台虚拟机迁移方法,包括:

[0011] 从源架构可执行文件中的二进制编码中截取一段,得到部分二进制代码;并得到所述部分二进制代码在源架构机器上运行的第一运行数据;

[0012] 针对所述异构虚拟化平台的N个异构芯片系统,翻译并运行所述部分二进制代码,得到所述部分二进制代码在所述N个异构芯片系统上运行的N个第二运行数据;

[0013] 基于所述第一运行数据和所述N个第二运行数据,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;其中,所述多个翻译流程包括静态翻译流程和动态翻译流程,所述静态翻译流程包括由静态翻译器实现的静态翻译过程,所述动态翻译流程包括由静态翻译器和动态翻译执行器实现的动静结合的翻译过程;

[0014] 基于所述目标架构进制和所述目标翻译流程,通过所述静态翻译器、所述动态翻译执行器和解释器中的一个或多个翻译所述源架构可执行文件,得到用于在目标架构机器上运行的可执行文件。

[0015] 为达上述目的,本申请第二方面实施例提出了一种异构虚拟化平台虚拟机迁移系统,包括:

[0016] 所述系统包括指令选择器、静态翻译器、动态翻译执行器和解释器,所述指令选择器,用于:

[0017] 从源架构可执行文件中的二进制编码中截取一段,得到部分二进制代码;并得到所述部分二进制代码在源架构机器上运行的第一运行数据;

[0018] 针对所述异构虚拟化平台的N个异构芯片系统,翻译并运行所述部分二进制代码,得到所述部分二进制代码在所述N个异构芯片系统上运行的N个第二运行数据;

[0019] 基于所述第一运行数据和所述N个第二运行数据,根据所述N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;其中,所述多个翻译流程包括静态翻译流程和动态翻译流程,所述静态翻译流程包括由静态翻译器实现的静态翻译过程,所述动态翻译流程包括由静态翻译器和动态翻译执行器实现的动静结合的翻译过程;

[0020] 基于所述目标架构进制和所述目标翻译流程,通过所述静态翻译器、所述动态翻译执行器和解释器中的一个或多个翻译所述源架构可执行文件,得到用于在目标架构机器上运行的可执行文件。

[0021] 为达上述目的,本申请第三方面实施例提出了一种电子设备,包括:处理器,以及与所述处理器通信连接的存储器;所述存储器存储计算机执行指令;所述处理器执行所述存储器存储的计算机执行指令,以实现第一方面所述的方法。

[0022] 为达上述目的,本申请第四方面实施例提出了一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,所述计算机执行指令被处理器执行时用于实现第一方面所述的方法。

[0023] 为达上述目的,本申请第五方面实施例提出了一种计算机程序产品,包括计算机程序,该计算机程序被处理器执行时实现第一方面所述的方法。

[0024] 本申请提供的异构虚拟化平台虚拟机迁移方法及系统,根据截取的源架构可执行文件的部分二进制代码在各异构芯片系统的运行数据,选择目标架构进制以及目标翻译流程;基于目标架构进制和目标翻译流程,翻译源架构可执行文件;再根据翻译效率判断当前的目标架构进制和目标翻译流程是否能够得到理想的翻译结果,在翻译结果不理想的情况下,调整目标架构进制和目标翻译流程重新进行翻译,得到目标架构系统的可执行文件;从而提高异构虚拟化平台虚拟机迁移效果。

[0025] 本申请附加的方面和优点将在下面的描述中部分给出,部分将从下面的描述中变得明显,或通过本申请的实践了解到。

附图说明

[0026] 本申请上述的和/或附加的方面和优点从下面结合附图对实施例的描述中将变得明显和容易理解,其中:

[0027] 图1为本申请实施例所提供的一种异构虚拟化平台虚拟机迁移方法的流程示意图;

[0028] 图2为本申请另一实施例所提供的一种异构虚拟化平台虚拟机迁移方法的流程示

意图;

[0029] 图3为本申请又一实施例所提供的一种异构虚拟化平台虚拟机迁移方法的流程示意图;

[0030] 图4为本申请又一实施例所提供的一种异构虚拟化平台虚拟机迁移方法的流程示意图;

[0031] 图5为本申请实施例所提供的一种异构虚拟化平台虚拟机迁移系统的框图;

[0032] 图6为本申请实施例所提供的一种电子设备的框图。

具体实施方式

[0033] 下面详细描述本申请的实施例,所述实施例的示例在附图中示出,其中自始至终相同或类似的标号表示相同或类似的元件或具有相同或类似功能的元件。下面通过参考附图描述的实施例是示例性的,旨在用于解释本申请,而不能理解为对本申请的限制。

[0034] 下面参考附图描述本申请实施例的异构虚拟化平台虚拟机迁移方法、装置及设备。

[0035] 图1为本申请实施例所提供的一种异构虚拟化平台虚拟机迁移方法的流程示意图。

[0036] 需要说明的是,本申请实施例的异构虚拟化平台虚拟机迁移方法的执行主体为本申请实施例的异构虚拟化平台虚拟机迁移装置,该异构虚拟化平台虚拟机迁移装置可被配置于电子设备中,以使该电子设备可以执行异构虚拟化平台虚拟机迁移功能。

[0037] 如图1所示,该异构虚拟化平台虚拟机迁移方法包括以下步骤:

[0038] 步骤101,从源架构可执行文件中的二进制编码中截取一段,得到部分二进制代码;并得到部分二进制代码在源架构机器上运行的第一运行数据;

[0039] 在一些实施例中,第一运行数据包括部分二进制代码在源架构机器上运行的第一执行时间。

[0040] 示例性的,源架构可执行文件为X86可执行文件,从X86可执行文件的X86二进制编码中截取一段二进制代码,将该段二进制代码在X86机器上运行,得到第一运行数据。

[0041] 步骤102,针对异构虚拟化平台的N个异构芯片系统,翻译并运行部分二进制代码,得到部分二进制代码在N个异构芯片系统上运行的N个第二运行数据。

[0042] 作为一种实现方式,针对异构虚拟化平台的N个异构芯片系统,通过静态翻译器翻译部分二进制代码,得到分别对应N个异构芯片系统的N个可执行代码;令N个可执行代码分别在N个异构芯片系统上运行,得到部分二进制代码在N个异构芯片系统上运行的N个第二运行数据。

[0043] 在一些实施例中,第二运行数据包括部分二进制代码在第n个异构芯片系统上运行的第二执行时间、CPU资源占用率、内存资源占用率和存储资源占用值;其中, $1 \leq n \leq N$ 。

[0044] 示例性的,开采研究院包括N个异构芯片系统,例如LoognArch、MIPS、SW64、RISC-V等。

[0045] 步骤103,基于第一运行数据和N个第二运行数据,根据N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;其中,多个翻译流程包括静态翻译流程和动态翻译流程,静态翻译流程包括由静态翻译器实现的静态翻译过程,动态翻

译流程包括由静态翻译器和动态翻译执行器实现的动静结合的翻译过程。

[0046] 作为一种实现方式,如图2所示,确定目标架构进制以及目标翻译流程的方法;包括:

[0047] 步骤201,基于第一执行时间和部分二进制代码在第n个异构芯片系统上运行的第二执行时间,得到部分二进制代码在第n个异构芯片系统上运行的执行效率。

[0048] 步骤202,基于部分二进制代码在第n个异构芯片系统上运行的执行效率、CPU资源占用率、内存资源占用率和存储资源占用值,得到N个异构芯片系统的资源值。

[0049] 步骤203,基于N个异构芯片系统的资源值,根据N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程。

[0050] 作为一种实现方式,执行效率通过如下公式获取:

[0051] $1 - (t_2 - t_1) / t_1$

[0052] 其中,t1为第一执行时间,t2为第二执行时间。

[0053] 在一些实施例中,N个异构芯片系统的资源值通过如下公式(1)计算得到:

$$[0054] \quad V_n = \frac{E_n - \frac{\lambda_1 * C_n + \lambda_2 * M_n + \lambda_3 * S_n}{\lambda_1 + \lambda_2 + \lambda_3}}{\frac{\lambda_1 * C_n + \lambda_2 * M_n + \lambda_3 * S_n}{\lambda_1 + \lambda_2 + \lambda_3}} \quad (1)$$

[0055] 其中,Vn为第n个异构芯片系统的资源值, λ_1 、 λ_2 、 λ_3 均为调节参数,根据虚拟化平台的CPU、内存、存储的实际情况及重要程度进行调节;En为在第n个异构芯片系统上运行的执行效率,Cn为在第n个异构芯片系统上运行的CPU资源占用率,Mn为在第n个异构芯片系统上运行的内存资源占用率,Sn为在第n个异构芯片系统上运行的存储资源占用值。

[0056] 在一些实施例中,如图3所示,基于第n个异构芯片系统的资源值,根据N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程的方法;包括:

[0057] 步骤301,基于N个异构芯片系统的资源值,确定资源值最大的目标异构芯片系统。

[0058] 步骤302,根据目标异构芯片系统的架构,确定目标架构进制。

[0059] 步骤303,当目标异构芯片系统的资源值大于资源值上限阈值时,选择静态翻译流程作为目标翻译流程。

[0060] 步骤304,当目标异构芯片系统的资源值小于等于资源值上限阈值且大于等于资源值下限阈值时,选择动态翻译流程作为目标翻译流程。

[0061] 需要说明的是,翻译流程还包括解释器翻译流程,解释器翻译流程包括解释器直接运行过程。

[0062] 示例性的,从V1...Vn中,选择Max(Vn)的异构芯片系统进行优先分配,例如龙芯架构系统的V值最高,则优先选择将X86可执行代码翻译成龙芯二进制代码,即将龙芯二进制作为目标架构进制。资源值上限阈值为1,资源值下限阈值为0,当0<目标异构芯片系统的V<=1时,选择动态翻译流程;当目标异构芯片系统的V>1时,选择静态翻译流程。

[0063] 这里需要说明的是,当Vn为负数时,说明该异构芯片系统资源耗尽,不能进行分配。

[0064] 步骤104,基于目标架构进制和目标翻译流程,通过静态翻译器、动态翻译执行器

和解释器中的一个或多个翻译源架构可执行文件,得到用于在目标架构机器上运行的可执行文件。

[0065] 作为一种实现方式,如图4所示,基于目标架构进制和目标翻译流程,通过静态翻译器、动态翻译执行器和解释器中的一个或多个翻译源架构可执行文件,得到用于在目标架构机器上运行的可执行文件的方法;包括:

[0066] 步骤401,当目标翻译流程为静态翻译流程时,通过静态翻译器翻译源架构可执行文件,得到翻译后的目标可执行文件。

[0067] 步骤402,当目标翻译流程为动态翻译流程时,通过静态翻译器和动态翻译执行器以动静结合方式翻译源架构可执行文件,得到翻译后的目标可执行文件;

[0068] 步骤403,获取静态翻译器的翻译效率。

[0069] 步骤404,基于翻译效率和预设阈值,确定新的目标架构进制、新的目标翻译流程以及是否将翻译后的目标可执行文件作为用于在目标架构机器上运行的可执行文件中的一个或者多个。

[0070] 作为一种实现方式,获取静态翻译器的翻译效率的方法,包括:获取静态翻译器的翻译量为 T ,总代码量为 C ,计算 T/C ,得到翻译效率。

[0071] 在一些实施例中,通过静态翻译器和动态翻译执行器以动静结合方式翻译源架构可执行文件,得到翻译后的目标可执行文件的方法;包括:通过静态翻译器对源架构可执行文件中的二进制编码进行离线翻译和深度优化,得到静态翻译结果;并将静态翻译结果传递至动态翻译执行器;通过动态翻译执行器装入源架构可执行文件中的二进制编码和静态翻译结果;对源架构可执行文件中的二进制编码中未被翻译的部分代码进行在线翻译和优化,并将在线翻译和优化的结果合并至静态翻译结果,得到翻译后的目标可执行文件。

[0072] 在一些实施例中,预设阈值包括第一效率阈值 a_1 和第二效率阈值 a_2 ,基于翻译效率和预设阈值,确定新的目标架构进制、新的目标翻译流程以及是否将翻译后的目标可执行文件作为用于在目标架构机器上运行的可执行文件中的一个或者多个的方法;包括:

[0073] 当翻译效率小于等于第一效率阈值时,通过解释器直接运行源架构可执行文件,得到用于在目标架构机器上运行的可执行文件;

[0074] 当翻译效率大于第一效率阈值且小于第二效率阈值、同时目标翻译流程为动态翻译流程时,则通过静态翻译器重新翻译源架构可执行文件,得到翻译后的目标可执行文件并返回重新获取静态翻译器的翻译效率;

[0075] 当翻译效率大于第一效率阈值且小于第二效率阈值、同时目标翻译流程为静态翻译流程时,基于第一运行数据和第二运行数据,返回重新从 N 个异构芯片系统中确定新的目标架构进制以及目标翻译流程;

[0076] 当翻译效率大于第二效率阈值时,将翻译后的目标可执行文件作为用于在目标架构机器上运行的可执行文件。

[0077] 可以理解为,当 $T/C < a_1$ 时,说明当前选择的目标翻译流程对于源架构二进制代码的翻译效果很差,大部分代码翻译失败,需要直接运行解释器运行该源架构二进制代码。当 $a_1 < T/C \leq a_2$ 时,如果目标翻译流程为动态翻译流程,则直接通过静态翻译器重新翻译源架构二进制代码;如果目标翻译流程为静态翻译流程,则返回重新选择 V 值低于当前目标架构系统 V 值的较高系统重新进行目标翻译流程。当 $T/C > a_2$ 时,说明当前目标翻译流程翻译得到

的代码较为理想,可以在目标机器上迁移使用。

[0078] 本申请实施例的异构虚拟化平台虚拟机迁移方法,根据截取的源架构可执行文件的部分二进制代码在各异构芯片系统的运行数据,选择目标架构进制以及目标翻译流程;基于目标架构进制和目标翻译流程,翻译源架构可执行文件;再根据翻译效率判断当前的目标架构进制和目标翻译流程是否能够得到理想的翻译结果,在翻译结果不理想的情况下,调整目标架构进制和目标翻译流程重新进行翻译,得到目标架构系统的可执行文件;从而提高异构虚拟化平台虚拟机迁移效果。

[0079] 为了清楚的说明上述实施例,现通过具体的示例进行说明。图5为本申请实施例所提供的一种异构虚拟化平台虚拟机迁移系统的框图。如图5所示,本申请的异构虚拟化平台虚拟机迁移方法应用于异构虚拟化平台虚拟机迁移系统,该异构虚拟化平台虚拟机迁移系统包括指令选择器501、静态翻译器502、动态翻译执行器503和解释器504,指令选择器501,用于:

[0080] 从源架构可执行文件中的二进制编码中截取一段,得到部分二进制代码;并得到部分二进制代码在源架构机器上运行的第一运行数据;

[0081] 针对异构虚拟化平台的N个异构芯片系统,翻译并运行部分二进制代码,得到部分二进制代码在N个异构芯片系统上运行的N个第二运行数据;

[0082] 基于第一运行数据和N个第二运行数据,根据N个异构芯片系统的架构确定目标架构进制以及从多个翻译流程中选择目标翻译流程;其中,多个翻译流程包括静态翻译流程和动态翻译流程,静态翻译流程包括由静态翻译器502实现的静态翻译过程,动态翻译流程包括由静态翻译器502和动态翻译执行器503实现的动静结合的翻译过程;

[0083] 基于目标架构进制和目标翻译流程,通过静态翻译器502、动态翻译执行器503和解释器504中的一个或多个翻译源架构可执行文件,得到用于在目标架构机器上运行的可执行文件。

[0084] 下面以X86源二进制代码为例进行说明。

[0085] 为了结合静态翻译器502和动态翻译执行器503的优点并避免它们的不足,构造动静结合的二进制翻译系统。此二进制翻译系统包含一个静态翻译器502和动态翻译执行器503,工作时首先由指令选择器501选择翻译的目标代码的格式,并根据性能选择是经过动态翻译执行器503翻译还是静态翻译器502直接翻译。其中,静态翻译器502对X86可执行文件进行离线翻译和深度优化,然后将翻译出的静态翻译结果交给动态翻译执行器503。动态翻译执行器503将X86的源文件和静态翻译结果一并装入,如果在执行过程中发现仍有未被翻译过的X86代码,则调用动态翻译器进行在线翻译和优化,将在线翻译和优化的结果合并至静态翻译结果一并利用。

[0086] 其中,静态翻译器502的作用是对X86源二进制代码进行翻译、优化,产生目标虚拟机上可执行代码,并以文件的形式存储。且静态翻译器502可以选择借助动态翻译执行器503的动态翻译信息对X86源二进制代码进行翻译、优化。静态翻译器502的作用是在程序未执行之前,将X86二进制进行翻译,生成静态翻译结果。该静态翻译结果包括翻译后的目标进制代码以及一些辅助信息。静态翻译器502执行的是离线翻译,离线翻译的时间不占用运行时间,翻译得到的静态翻译结果可以供动态翻译执行器503多次使用;因此静态翻译器502可以实现一些力度更大的优化,生成更加高效的目标架构代码。但静态翻译器502也有

一些局限性,例如间接跳转和间接调用的目标不能识别,不能处理代码自修改程序等,这些必须依靠动态翻译执行器503去解决。

[0087] 静态翻译器502所实现的主要优化包括局部寄存器分配(LRA)、全局寄存器分配(GRA)和扩展基本块优化(EB0),其中,局部寄存器分配指的是在基本块内的寄存器分配算法;全局寄存器分配指的是跨越基本块的寄存器分配算法;扩展基本块优化指的是跨越基本块的许多编译常规优化,如常量传播,死代码删除等。

[0088] 为了解决静态翻译器502存在不能处理间接跳转目标的局限性,可以通过语义提升,对一类间接跳转的目标成功实现了识别,在很大程度上克服了静态翻译器502的这种局限,也提高了系统迁移的性能。

[0089] 在一些实施例中,静态翻译器502的结构如图3所示,包括文件装入单元、翻译单元、优化单元和翻译结果输出单元,其中,文件装入单元用于将X86源二进制代码和动态翻译执行器503生成的翻译信息装入内存,然后由翻译单元进行翻译,翻译出的中间结果交由优化单元进行优化(翻译单元也会在翻译过程中进行一定的优化),最后由翻译结果输出单元将翻译单元和优化单元的输出结果按照一定的格式输出,得到静态翻译结果,传递给动态翻译执行器503使用。

[0090] 其中,动态翻译执行器503的作用是利用静态翻译器502的静态翻译结果,通过翻译、本地码执行和解释等手段执行目标进制程序。程序的执行单元为一个Block块,即一组线形的X86指令代码,直到遇到跳转指令结束。在执行过程中进行动态翻译,记录动态翻译信息结果供静态翻译器502使用。异构虚拟化平台虚拟机迁移系统的动态部分的输入是静态部分生成的文件和X86的二进制源码。动态部分以源码的Block为单位,解释或翻译、执行生成的本地码。

[0091] 动态翻译执行器503的功能是对X86源二进制文件,充分利用静态翻译器502的翻译结果,并辅助解释和动态翻译的方法,实现X86可执行文件在目标机器上正确执行的效果。动态翻译执行器503的系统构成如图所示

[0092] 在一些实施例中,动态翻译执行器503包括动态总控模块、文件装载模块、命令行参数分析模块、反汇编模块、本地码执行环境模块、解释模块、翻译模块和优化模块,其中,动态总控模块负责协调各模块的工作,文件装载模块负责装载X86可执行文件以及静态翻译器502的翻译结果,命令行参数分析模块负责分析程序启动时的命令行参数,获取被翻译程序信息,反汇编模块负责对X86可执行代码进行反汇编,并形成动态翻译或解释的基本单元,本地码执行环境模块负责构造本地码运行的上下文环境,包括寄存器映射,本地码装载执行,上下文切换,内存管理等。解释模块负责对反汇编后的X86可执行代码的解释执行,翻译模块负责对反汇编后的X86可执行代码的翻译,优化模块包括为了提高二进制翻译系统的性能采取的优化措施。下面对主要模块的算法和实现进行详细描述。

[0093] 动态总控模块是动态执行的总控模块,它控制对X86 Block的解释、翻译以及本地码的执行。该模块从X86入口地址开始,模拟X86程序执行语义,对于存在本地码的基本块,直接执行本地码,对于不存在本地码的基本块,先翻译后执行。直到模拟的X86程序结束,本模块结束。二进制翻译系统中动态翻译部分中动态执行总控模块的执行流程,把总控模块划分为4个主要的子模块,即基本块划分模块,解释器504模块,动态翻译模块,本地码执行模块。动态总控模块包括如下实现流程:

[0094] 1) 把入口地址赋值给pc (pc是X86代码的程序计数器,也就是X86代码的虚地址,本模块将要翻译执行的X86地址)。根据pc内容构建information。(information中的内容将在翻译或者解释的时候用到)。

[0095] 2) 判断pc对应的X86地址是否已经有本地码存在。如果存在本地码,转7) 本地码执行。

[0096] 3) 构建X86 Block(Block划分)。

[0097] 4) 判断以pc为入口的Block是需要解释执行还是翻译执行。如果解释模拟该Block,到5),否则转6) 本地码翻译子模块。

[0098] 5) 解释器504模拟X86指令,更新pc,转8)。

[0099] 6) 翻译pc入口的Block,并且建立和已有本地码的链接。

[0100] 7) 本地码执行,执行完毕后返回一些构建information的信息,包括更新pc。

[0101] 8) 判断是否X86程序结束,如果结束,本模块结束。否则记录间接目标地址;转2)。

[0102] 反汇编模块负责划分Block并对Block进行反汇编,记录信息到IR1中。block结束的指令有:branch、jmp^间接跳转、间接调用、ret和halt。反汇编模块实现如下流程:

[0103] 1) 将block->entry_vaddr赋给pc。

[0104] 2) 对pc所指的指令进行反汇编。

[0105] 3) 取下一条指令。

[0106] 4) 判断该指令的类型是否是Block结束指令;否则,转2)。

[0107] 5) 将对该block反汇编出的IR1存入该block的irl缓存中。

[0108] 6) 设置该block的irl num为IR1数目,irl addr为irl缓存地址,标记该block的flag为已被反汇编。

[0109] 反汇编模块主要用于反汇编一个给定的动态的翻译单元。它以该翻译单元的入口pc作为参数,对该翻译单元进行反汇编,反汇编结果存入IR1中,并记录block信息。

[0110] 动态翻译执行器503执行的X86文件的指令划分成基本块的形式,反汇编模块需要逐条反汇编基本块内每一条指令,将反汇编的结果记录在X86指令的中间表示中,直到识别出基本块最后一条控制转移指令,动态反汇编的一次执行过程结束。

[0111] 解释模块包括两部分组成,一部分是运行部分,在反汇编模块完成操作,将X86的二进制代码表示成IR1中间表示形式之后所执行的模块,此模块是逐条的模拟解释X86指令的执行;另一部分是初始化部分,在执行对指令的解释之前对虚拟执行环境的初始化。其中,运行部分由一个总控函数,和一系列指令解释函数组成,总控函数是个总调度器,循环的解释每一条指令,具体的指令的解释调用每一个具体的指令解释函数(几乎所有的X86指令都对应着一个指令解释函数)。对于每一条指令的解释通常是预先分析指令的含义,然后用同等语义的指令去解释并且同时执行。初始化部分只是完成对虚拟运行环境的初始化。

[0112] 翻译模块有两个子模块,分别是标志位分析模块和指令翻译模块,其中,标志位分析模块工作在指令翻译之前,可以很好的提高翻译出来的IR2指令序列的效率;作用是对反汇编模块生成的X86源程序进行翻译,生成具同样语义的IR2指令序列。同时,组织动态寄存器分配、标志位计算等与翻译密切相关的模块同时工作。该模块的最基本单位是一系列指令翻译函数,每个翻译函数对应一个操作语义相同的X86指令。模块从总入口函数Translate()开始,经过条件判断翻译标记、根据编译选项选择加载相关功能等步骤,最后

间接的调用这一系列的X86指令翻译函数来实现源程序的语义。在动态翻译过程中,flag pattern、标志位计算、寄存器动态分配等重要的相关模块同时工作,其中flag pattern模块和标志位计算模块可以有效的提高翻译出的IR2指令的效率。

[0113] 本地码执行环境模块负责构造本地码运行的上下文环境。在BT控制器发现某个Block已经被翻译过,调用本模块实现对该Block启动执行,从BT控制器转入本地码执行和从本地码执行转入控制器,都需要进行上下文的切换。由于翻译中采用本地码链接技术,本地码一次执行可能有多个Block被执行。

[0114] 为了实现上述实施例,本申请还提出一种电子设备。请参见图6,图6是本申请实施例提供的电子设备的框图。如图6所示,电子设备600包括:处理器601,以及与处理器601通信连接的存储器602;存储器602存储计算机执行指令;处理器601执行存储器存储的计算机执行指令,以实现执行前述实施例所提供的方法。

[0115] 为了实现上述实施例,本申请还提出一种计算机可读存储介质,计算机可读存储介质中存储有计算机执行指令,所述计算机执行指令被处理器执行时用于实现前述实施例所提供的方法。

[0116] 为了实现上述实施例,本申请还提出一种计算机程序产品,包括计算机程序,该计算机程序被处理器执行时实现前述实施例所提供的方法。

[0117] 本申请中所涉及的用户个人信息的收集、存储、使用、加工、传输、提供和公开等处理,均符合相关法律法规的规定,且不违背公序良俗。

[0118] 需要说明的是,来自用户的个人信息应当被收集用于合法且合理的用途,并且不在这些合法使用之外共享或出售。此外,应在收到用户知情同意后进行此类采集/共享,包括但不限于在用户使用该功能前,通知用户阅读用户协议/用户通知,并签署包括授权相关用户信息的协议/授权。此外,还需采取任何必要步骤,保卫和保障对此类个人信息数据的访问,并确保有权访问个人信息数据的其他人遵守其隐私政策和流程。

[0119] 本申请预期可提供用户选择性阻止使用或访问个人信息数据的实施方案。即本公开预期可提供硬件和/或软件,以防止或阻止对此类个人信息数据的访问。一旦不再需要个人信息数据,通过限制数据收集和删除数据可最小化风险。此外,在适用时,对此类个人信息去除个人标识,以保护用户的隐私。

[0120] 在前述各实施例描述中,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申请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。

[0121] 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。在本申请的描述中,“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。

[0122] 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定制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,并且本申请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,这应被本申请的实施例所属技术领域的技术人员所理解。

[0123] 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),可擦除可编程只读存储器(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。

[0124] 应当理解,本申请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。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它们的组合来实现: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。

[0125] 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。

[0126] 此外,在本申请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。

[0127] 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申请的实施例,可以理解的是,上述实施例是示例性的,不能理解为对本申请的限制,本领域的普通技术人员在本申请的范围内可以对上述实施例进行变化、修改、替换和变型。

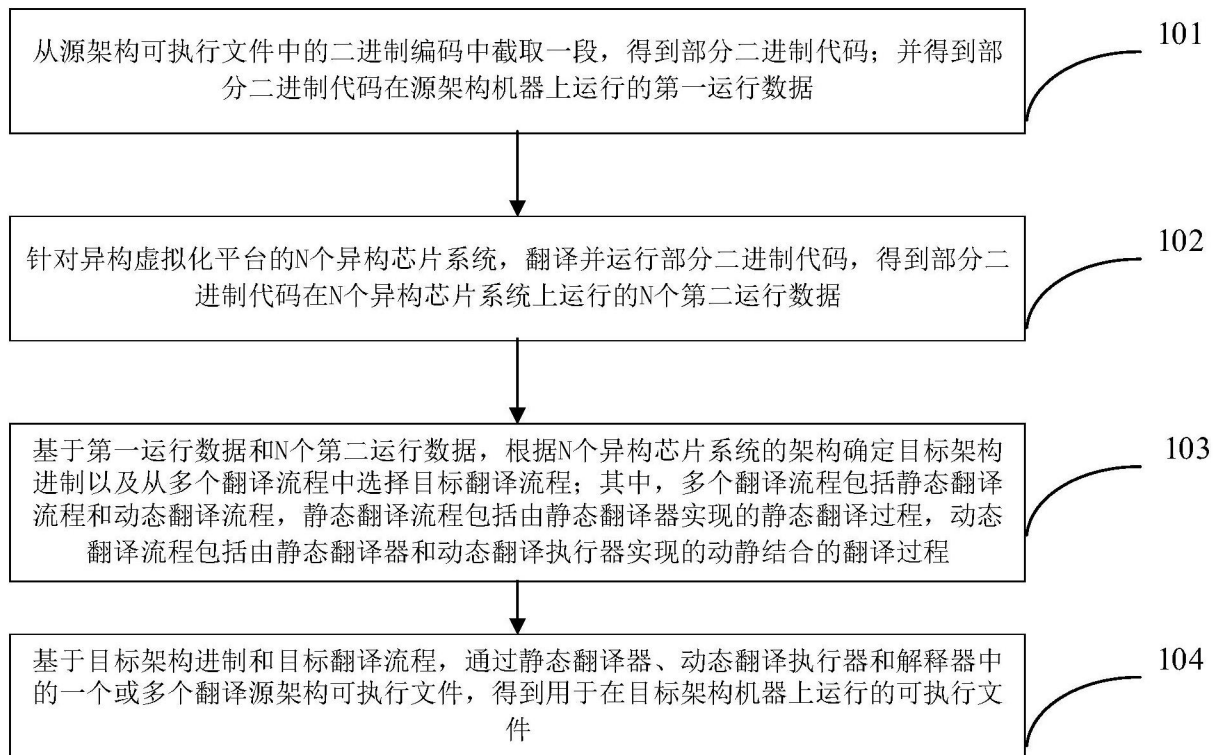


图1

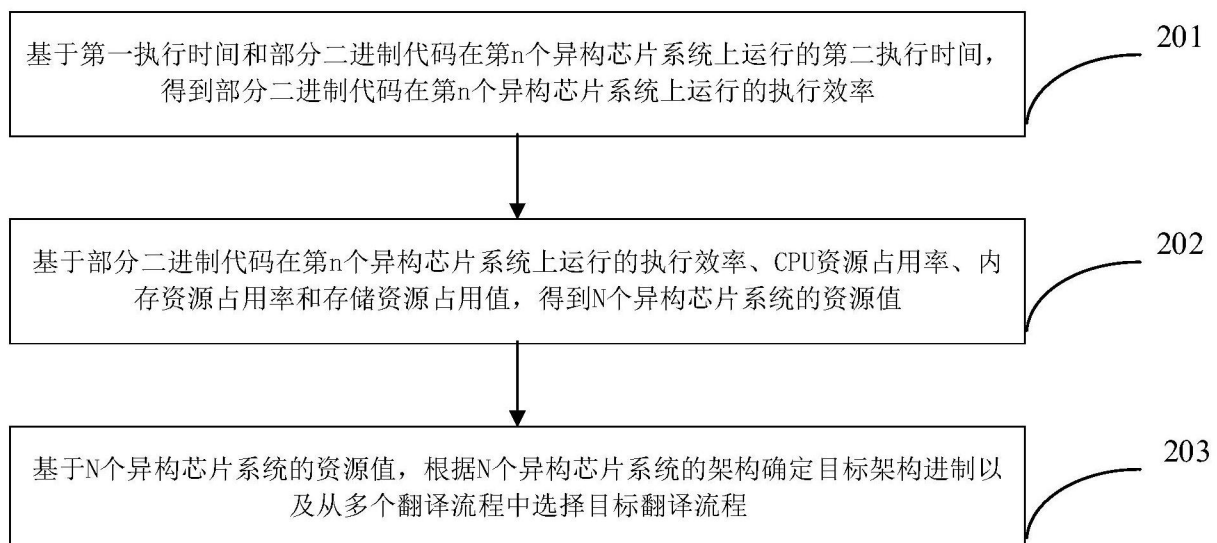


图2

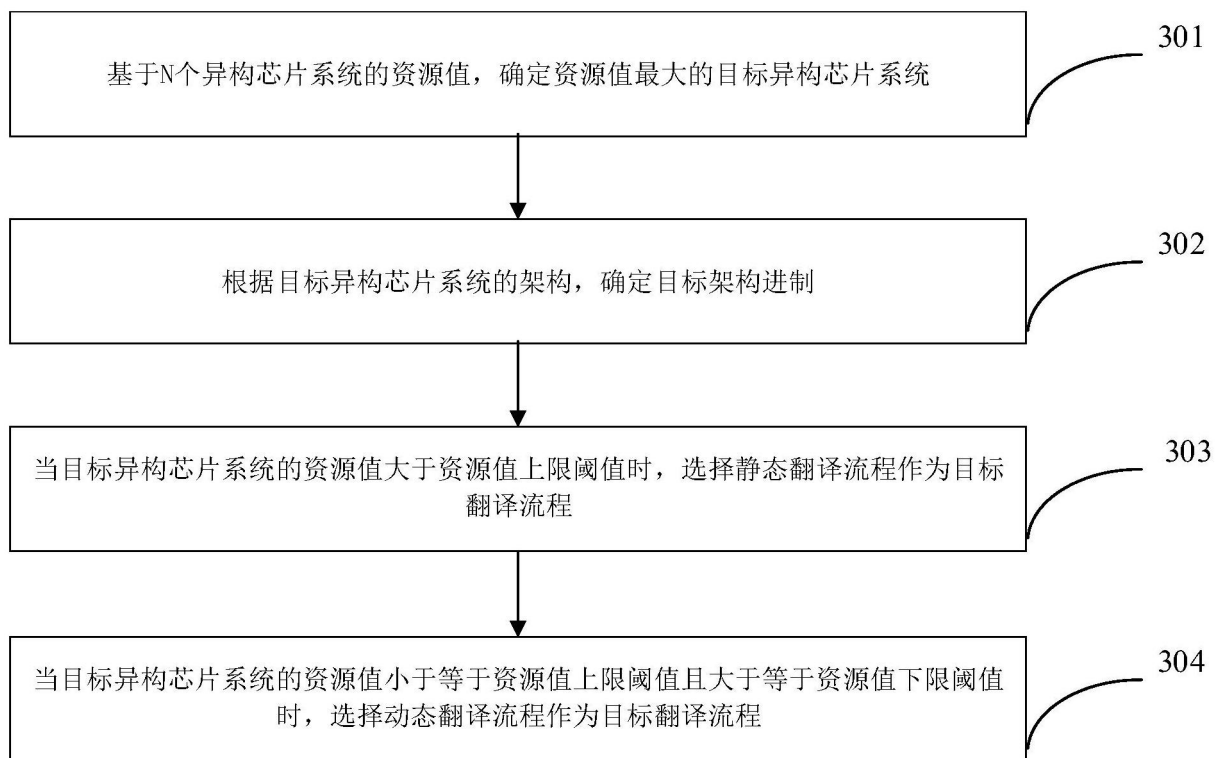


图3

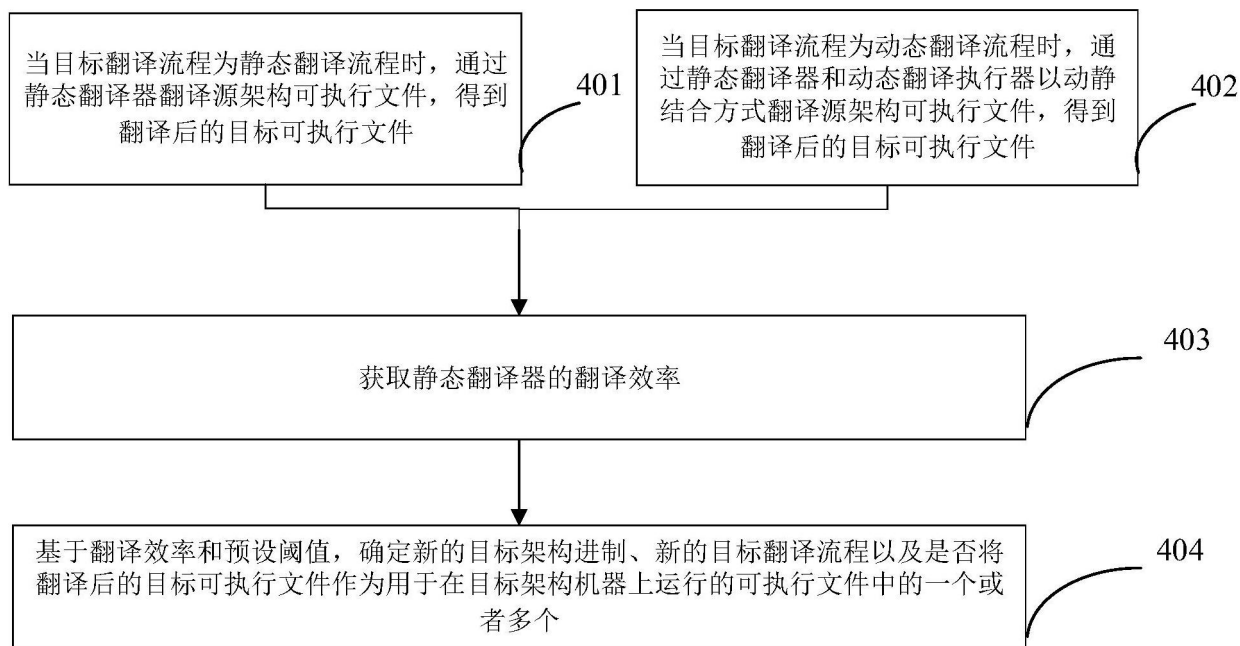


图4

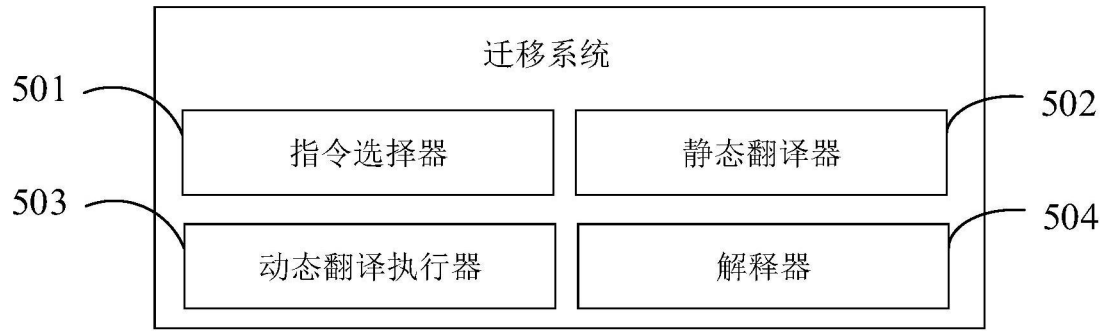


图5

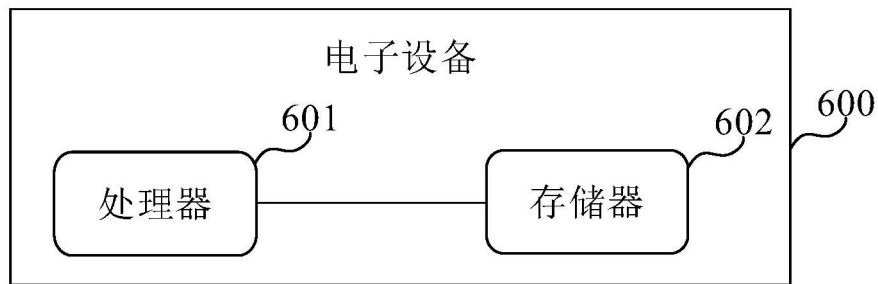


图6