

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2024 年 8 月 22 日 (22.08.2024)



(10) 国际公布号
WO 2024/169182 A1

(51) 国际专利分类号:

G06F 8/20 (2018.01)

(21) 国际申请号:

PCT/CN2023/120630

(22) 国际申请日:

2023 年 9 月 22 日 (22.09.2023)

(25) 申请语言:

中文

(26) 公布语言:

中文

(30) 优先权:

202310126681.5 2023年2月16日 (16.02.2023) CN

(71) 申请人: 华为云计算技术有限公司 (HUAWEI

CLOUD COMPUTING TECHNOLOGIES CO., LTD.)

[CN/CN]; 中国贵州省贵阳市贵安新区黔中大道交

兴功路华为云数据中心, Guizhou 550025 (CN)。

(72) 发明人: 高战 (GAO, Zhan); 中国贵州省贵阳市贵安

新区黔中大道交兴功路华为云数据中心, Guizhou

550025 (CN)。张五贵 (ZHANG, Wugui); 中国贵州

省贵阳市贵安新区黔中大道交兴功路华为云数据

中心, Guizhou 550025 (CN)。贺佳君 (HE, Jiajun);

中国贵州省贵阳市贵安新区黔中大道交兴功路

华为云数据中心, Guizhou 550025 (CN)。

(74) 代理人: 深圳市深佳知识产权代理事务所 (普

通合伙) (SHENPAT INTELLECTUAL PROPERTY

AGENCY); 中国广东省深圳市罗湖区南湖街

道春风路庐山大厦 B 座 18C2、18D、18E、

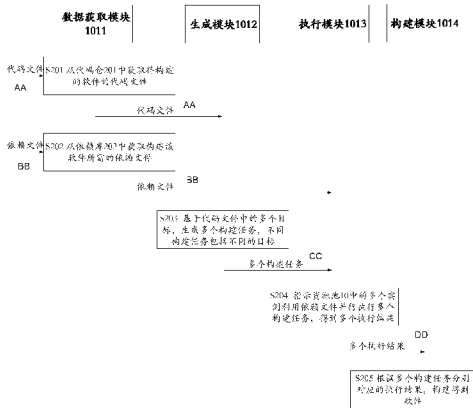
18E2, Guangdong 518001 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家

保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,

(54) Title: METHOD FOR SOFTWARE CONSTRUCTION, APPARATUS, AND RELATED DEVICE

(54) 发明名称: 构建软件的方法、装置及相关设备



(57) Abstract: Provided is a method for software construction, comprising: a first instance acquiring a code file of software to-be-constructed from a code repository, and acquiring from a dependency library a dependency file required for constructing the software; the first instance generating, on the basis of a plurality of targets in the code file, a plurality of construction tasks, different construction tasks comprising different targets, and instructing a plurality of instances in a resource pool to execute the plurality of construction tasks in parallel by utilizing the dependency file to obtain execution results corresponding to the plurality of construction tasks, so that the first instance constructs and obtains the software according to the execution results. In this way, in the process of software construction, the first instance executes the construction process in parallel on the plurality of construction tasks comprising the different targets, and compared with a mode of constructing targets one by one on a single machine, the construction efficiency for the plurality of targets can be effectively improved, thereby effectively improving the overall efficiency of software construction.

(57) 摘要: 提供了一种构建软件的方法, 包括: 第一实例从代码仓库中获取待构建的软件的代码文件, 并从依赖库中获取构建软件所需的依赖文件; 第一实例基于代码文件中的多个目标, 生成多个构建任务, 不同的构建任务包括不同目标, 并指示资源池中的多个实例利用依赖文件并行执行多个构建任务, 得到多个构建任务分别对应的执行结果, 从而第一实例根据多个执行结果, 构建得到软件。如此, 在构建软件的过程中, 第一实例对包括不同目标的多个构建任务并行执行构建过程, 这相比于在单机上逐个构建目标的方式, 可以有效提升针对多个目标的构建效率, 以此可以有效提升构建软件的整体效率。

BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

构建软件的方法、装置及相关设备

本申请要求于 2023 年 2 月 16 日提交中国专利局、申请号为 202310126681.5、发明名称为“构建软件的方法、装置及相关设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机技术领域，尤其涉及一种构建软件的方法、装置及相关设备。

背景技术

构建软件，是指将源代码通过编译、链接、打包等环节，转换为软件包的过程，以此实现软件开发。实际应用场景中，随着软件项目需求的不断增加，软件项目的代码量也会随之增加，这使得在单机上构建软件的耗时较高，如可能达到数个小时或者数天之久，从而严重影响软件开发效率。

因此，如何提高构建软件的效率，成为亟需解决的重要技术问题。

发明内容

有鉴于此，本申请实施例提供了一种构建软件的方法，用于缩短构建软件所需的耗时，以此提高软件开发效率。本申请还提供了对应的装置、计算设备集群、计算机可读存储介质以及计算机程序产品。

第一方面，本申请实施例提供了一种构建软件的方法，该方法应用于资源池中的第一实例，该第一实例例如可以是资源池中的主机、虚拟机、容器等；在构建软件的过程中，第一实例从代码仓中获取待构建的软件的代码文件，并从依赖库中获取构建该软件所需的依赖文件，如构建软件所需的 Jar 包等；然后，第一实例基于该代码文件中的多个目标，生成多个构建任务，其中，不同的构建任务可以包括不同的所要构建的目标（target），并指示该资源池中的多个实例利用依赖文件并行执行多个构建任务，得到该多个构建任务分别对应的执行结果，该多个实例可以包括第一实例，也可以不包括第一实例，从而第一实例根据该多个构建任务分别对应的执行结果，构建得到软件。

如此，在构建软件的过程中，第一实例对包括不同目标的多个构建任务并行执行构建过程，这相比于在单机上逐个构建目标的方式，可以有效提升针对该多个目标的构建效率，以此可以有效提升构建软件的整体效率。

在一种可能的实施方式中，第一实例在获取代码文件时，具体可以是对代码仓中所要构建的软件的代码文件进行分布式缓存，以此可以提高第一实例获取代码文件的效率，进而可以提高构建软件的整体效率。

可选地，第一实例也可以是通过一致性哈希、分布式文件存储或者增量更新等技术，对代码文件进行缓存，以此提高第一实例获取代码文件的效率。

在一种可能的实施方式中，第一实例在生成多个构建任务时，具体可以是确定代码文件中的编译控制文件，如 cmakelist 文件、ninja 文件等，该编译控制文件包括多个目标以及该多个目标的依赖信息；然后，第一实例根据该编译控制文件，为所要构建的软件生成多个构建任务。如此，第一实例可以根据编译控制文件实现多个构建任务的生成。

在一种可能的实施方式中，参与生成多个构建任务的多个目标，为编译控制文件中定义的全部目标；或者，参与生成多个构建任务的多个目标，为编译控制文件中满足预设条件的目标，该预设条件包括预设类型，costumer 类型、external project 类型等，或者预设条件包括预设标识，如指定的文件名称等。实际应用时，可以根据需要灵活选择编译控制文件中的部分或者全部目标来生成多个构建任务，以实现加速软件构建的目的。

在一种可能的实施方式中，所述第一实例输出配置界面，例如可以是向对外提供的客户端输出配置界面，该配置界面可以通过客户端呈现给用户，从而第一实例响应于用户在该配置界面上针对预设条件的配置操作，确定预设条件。如此，用户可以在配置界面上灵活选择需要参与构建加速的多个目标，以此可以提高软件构建过程中对于目标选择的灵活性。

在一种可能的实施方式中，第一实例在指示资源池中的多个实例利用依赖文件并行执行多个构建任务时，具体可以是按照预设规则，将多个构建任务分发给多个实例，该多个构建任务被多个实例并行执行，并且不同实例执行不同的构建任务，从而第一实例可以获取该多个实例反馈的多个构建任务分别对应的执行结果。如此，第一实例可以利用资源池的多个实例来实现加速构建多个目标，进而实现提高软件构建的整体效率。

在一种可能的实施方式中，第一实例在生成多个构建任务时，具体可以是根据多个实例的数量或多个目标的构建产物所属的目录，对编译控制文件中的多个目标进行分组，如将构建产物位于同一目录下的一个或者多个目标划入一组等，以此得到多个构建任务，其中，每个构建任务包括的目标为多个目标中的一组目标。如此，按照构建产物所属的目录对多个目标进行分组以及构建后，可以提高针对该多个目标的构建效率。

在一种可能的实施方式中，编译控制文件中还包括除参与生成多个构建任务的多个目标之外的其他目标，则，在软件构建过程中，第一实例还可以构建该编译控制文件中的其它目标，得到本地执行结果，并根据该本地执行结果以及多个构建任务分别对应的执行结果，构建得到软件。如此，第一实例可以与资源池中的其它多个实例共同完成软件的加速构建。

第二方面，本申请实施例提供一种构建软件的方法，该方法包括：获取待构建的软件的代码文件中的编译控制文件，如 `cmakelist` 文件、`ninja` 文件等，该编译控制文件包括多个目标以及该多个目标的依赖信息，并获取构建该软件所需的依赖文件；然后，根据该编译控制文件为该软件生成多个构建任务，该多个构建任务中的不同构建任务包括不同的目标；接着，并行执行该多个构建任务，得到多个构建任务分别对应的执行结果，并根据该多个构建任务分别对应的执行结果，构建得到软件。如此，在构建软件的过程中，通过对包括不同目标的多个构建任务并行执行构建过程，这相比于在单机上逐个构建目标的方式，可以有效提升针对该多个目标的构建效率，以此可以有效提升构建软件的整体效率。

第三方面，本申请实施例还提供了一种软件构建装置，该软件构建装置包括用于执行第一方面或第一方面任一种可能实现方式中的构建软件的方法的各个模块。

第四方面，本申请实施例还提供了一种软件构建装置，该软件构建装置包括用于执行第二方面所述实施方式中的构建软件的方法的各个模块。

第五方面，本申请提供一种计算设备集群，所述计算设备包括至少一个计算设备，所述至少一个计算设备包括至少一个处理器和至少一个存储器；所述至少一个存储器用于存储指令，所述至少一个处理器执行所述至少一个存储器存储的该指令，以使所述计算设备集群执行上述第一方面或第一方面任一种可能实现方式中的构建软件的方法，或者以使得所述计算设备集群执行上述第二方面实施方式中的构建软件的方法。需要说明的是，该存储器可以集成于处理器中，也可以是独立于处理器之外。所述至少一个计算设备还可以包括总线。其中，处理器通过总线连接存储器。其中，存储器可以包括可读存储器以及随机存取存储器。

第六方面，本申请提供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指令，当其在至少一个计算设备上运行时，使得所述至少一个计算设备执行上述第一方面或第一方面的任一种实现方式所述的方法，或者使得所述至少一个计算设备执行上述第二方面实施方式中的构建软件的方法。

第七方面，本申请提供了一种包含指令的计算机程序产品，当其在至少一个计算设备上运行时，使得所述至少一个计算设备执行上述第一方面或第一方面的任一种实现方式所述的方法，或者使得所述至少一个计算设备执行上述第二方面实施方式中的构建软件的方法。

本申请在上述各方面提供的实现方式的基础上，还可以进行进一步组合以提供更多实现方式。

附图说明

图1为本申请实施例提供的一示例性应用场景示意图；

图2为本申请实施例提供了一种构建软件的方法的流程示意图；

图3为本申请实施例提供了一种配置界面的示意图；

图4为本申请实施例提供了一种计算设备的结构示意图；

图5为本申请实施例提供了一种计算设备集群的结构示意图。

具体实施方式

下面将结合本申请中的附图，对本申请提供的实施例中的方案进行描述。

通常情况下，用户（如开发人员等）在进行软件开发的过程中，通常需要对编写的源代码依次执行编译（compiling）、链接（linking）、打包等过程，以此完成软件构建。其中，编译，是指利用编译程序将由源语言编写的源代码程序翻译成基于目标语言的程序代码的过程，如将由 C、Java 等高级语言编写的源代码翻译成基于机器语言的程序代码等。链接，是指将各种程序代码和数据部分收集起来并组合成为一个单一文件的过程。打包，是指将多个代码文件组合并压缩成一个文件的过程。

实际应用场景中，用户所开发软件的代码量较大，该软件的程序代码中所要构建的目标（target）的数量较多。其中，目标，具体可以是所要构建的文件名称，也可以是所要执行的操作名称，用于指示执行命令时的操作目的（也即指示要干什么）。因此，在针对程序代码中较多数量的目标逐个执行编译、链接以及打包等构建过程所需的耗时较高，导致在单机构建软件的耗时通常较高，从而影响用户开发软件的效率。

基于此，本申请实施例提供了一种构建软件的方法，用以缩短构建软件所需的耗时，以此提高软件开发效率。具体实现时，如图 1 所示，该方法可以由资源池 10 中的实例 101 执行，其中，资源池 10 中包括多个实例，图 1 中以包括实例 101、实例 102 以及实例 103 为例进行示例性说明。示例性地，资源池 10 中的各个实例，可以包括资源池 10 中的主机、虚拟机、容器中一种或者多种，或者可以是通过其它方式实现的实例，对此并不进行限定。

在构建软件的过程中，实例 101 从代码仓 201 获取待构建的软件的代码文件，并从依赖库 202 中获取构建该软件所需的依赖文件，然后，实例 101 基于获取的代码文件，生成多个构建任务，其中，不同构建任务包括代码文件中所要构建的不同目标。接着，实例 101 指示资源池 10 中的多个实例利用该依赖文件并行执行多个构建任务，得到多个构建任务分别对应的执行结果，其中，参与并行执行多个构建任务的多个实例，可以是实例 102 以及实例 103，进一步地，该多个实例还可以包括实例 101 等。从而，实例 101 根据该多个构建任务分别对应的执行结果，构建得到软件，以此完成软件开发。由于在构建软件的过程中，实例 101 对生成的多个构建任务并行执行构建过程，这相比于在单机上逐个构建目标的方式，可以有效提升针对该多个目标的构建效率，以此可以有效提升构建软件的整体效率。

示例性地，在构建软件的过程中，具体可以是实例 101 中的多个功能模块协同完成软件构建过程。如图 1 所示，实例 101 可以包括数据获取模块 1011、生成模块 1012、执行模块 1013 以及构建模块 1014。其中，数据获取模块 1011 用于获取待构建的软件的代码文件，并获取构建该软件所需的依赖文件，如从代码仓 201 获取代码文件，从依赖库 202 中获取依赖文件等，然后，数据获取模块 1011 可以将获取的代码文件提供给生成模块 1012，并将依赖文件提供给执行模块 1013。生成模块 1012 基于代码文件中的多个目标，生成多个构建任务，该多个构建任务中的不同构建任务包括不同的目标，并将该多个构建任务提供给执行模块 1013。执行模块 1013 指示资源池 10 中的多个实例利用该依赖文件并行执行该多个构建任务，得到相应的执行结果，并将该多个构建任务分别对应的执行结果提供给构建模块 1014，得到多个构建任务分别对应的执行结果，并将该多个执行结果提供给构建模块 1014。构建模块 1014 根据该多个构建任务分别对应的执行结果，构建得到所要开发的软件，以此完成软件开发。

进一步地，实例 101 还可以包括其它功能模块，如还可以包括配置模块等。其中，配置模块用于对实例 101 的构建过程进行相应的配置，如配置并行执行的构建任务的数量、配置被并行执行构建过程的目标所属的类型等。

实际应用时，实例 101 在构建得到软件后，还可以将该软件添加至软件库 203 中进行持久化存储，以便后续用户可以从该软件库 203 中获取到完成构建的软件。

作为一种示例，上述资源池 10、代码仓 201、依赖库 202 以及软件库 203 可以被部署于云端，用于为用户/租户提供构建软件的云服务。示例性地，该云端例如可以是公有云、私有云、边缘云等，并且，资源池 10 中的多个实例可以由云端的计算设备或者计算设备集群实现；代码仓 201、依赖库 202 以及软件库 203 可以由云端的存储设备或者存储设备集群实现。如此，能够提供构建软件的云服务。

作为另一种示例，上述资源池 10、代码仓 201、依赖库 202 以及软件库 203 可以被部署于本地，从

而可以为用户/租户提供本地的构建软件的服务。示例性地，资源池 10 中的多个实例可以通过用户侧的一个或者多个计算设备实现；代码仓 201、依赖库 202 以及软件库 203 可以通过用户侧的一个或者多个存储设备实现。

上述实例 101 可以通过软件实现，或者可以通过硬件实现。

实例 101 作为软件功能单元的一种举例，可以包括运行在计算实例上的代码。其中，计算实例可以包括主机、虚拟机、容器中的至少一种。进一步地，上述计算实例可以是一台或者多台。例如，实例 101 可以包括运行在多个主机/虚拟机/容器上的代码。需要说明的是，用于运行该代码的多个主机/虚拟机/容器可以分布在相同的区域（region）中，也可以分布在不同的 region 中。进一步地，用于运行该代码的多个主机/虚拟机/容器可以分布在相同的可用区（availability zone, AZ）中，也可以分布在不同的 AZ 中，每个 AZ 包括一个数据中心或多个地理位置相近的数据中心。其中，通常一个 region 可以包括多个 AZ。

同样，用于运行该代码的多个主机/虚拟机/容器可以分布在同一个虚拟私有云（virtual private cloud, VPC）中，也可以分布在多个 VPC 中。其中，通常一个 VPC 设置在一个 region 内，同一 region 内两个 VPC 之间，以及不同 region 的 VPC 之间跨区通信需在每个 VPC 内设置通信网关，经通信网关实现 VPC 之间的互连。

实例 101 作为硬件功能单元的一种举例，实例 101 可以包括至少一个计算设备，如服务器等。或者，实例 101 也可以是利用专用集成电路（application-specific integrated circuit, ASIC）实现、或可编程逻辑器件（programmable logic device, PLD）实现的设备等。其中，上述 PLD 可以是复杂程序逻辑器件（complex programmable logical device, CPLD）、现场可编程门阵列（field-programmable gate array, FPGA）、通用阵列逻辑（generic array logic, GAL）或其任意组合实现。

实例 101 包括的多个计算设备可以分布在相同的 region 中，也可以分布在不同的 region 中。实例 101 包括的多个计算设备可以分布在相同的 AZ 中，也可以分布在不同的 AZ 中。同样，实例 101 包括的多个计算设备可以分布在同一个 VPC 中，也可以分布在多个 VPC 中。其中，上述多个计算设备可以是服务器、ASIC、PLD、CPLD、FPGA 和 GAL 等计算设备的任意组合。

接下来，对构建软件的过程中各种非限定性的具体实施方式进行详细描述。

参阅图 2，为本申请实施例中一种构建软件的方法的流程示意图。该方法可以应用于上述图 1 所示的应用场景中，或者也可以是应用于其它可适用的应用场景中。为便于理解，下面以应用于图 1 所示的应用场景为例进行说明。图 2 所示的构建软件的方法具体可以包括：

S201：数据获取模块 1011 从代码仓 201 中获取待构建的软件的代码文件。

S202：数据获取模块 1011 从依赖库 202 中获取构建该软件所需的依赖文件。

本实施例中，实例 101 可以应用于使用构建框架的项目，该构建框架例如可以是 CMake 构建框架、Gradle 构建框架、Bazel 构建框架、Scons 构建框架中的任意一种，或者可以是其他可适用的构建框架等。相应的，使用构建框架的项目，例如可以是安卓开源项目（android open-source project, AOSP），或者是基于 AOSP 衍生的项目，如 Android、EMUI、MIUI、ColorOS、OriginOS、One UI、ZUI、氢 OS 等项目中的任意一种，或者可以是其他项目，如 OpenHarmony 项目、应用于汽车上的智能车机系统等。本实施例中，对于实例 101 所应用的构建框架以及项目并不进行限定。

在一种可能的实施方式中，实例 101 可以对外提供客户端，该客户端例如可以是网络浏览器，或者可以是部署于用户终端上的应用程序等。这样，用户可以通过客户端向实例 101 提供所要构建的软件的代码文件的标识，如代码文件的名称等，从而数据获取模块 1011 可以根据该标识，从代码仓 201 中获取相应的代码文件。

并且，用户可以通过客户端向实例 101 提供构建该软件所需的依赖文件的标识，从而数据获取模块 1011 可以根据该标识，从依赖库 202 中获取相应的依赖文件。其中，依赖文件，是指构建目标的过程中所需依赖的文件，如 Jar 包、软件开发工具包（Software Development Kit, SDK）等。实际应用时，针对基于不同程序语言所开发的软件，实例 101 可以利用不同的依赖文件完成软件构建。

进一步地，数据获取模块 1011 在获取代码文件时，具体可以是对代码仓 201 中待构建的软件的代

码文件进行分布式缓存, 以此实现加速获取代码文件, 以便于提高软件构建的整体效率。或者, 数据获取模块 101 也可以基于一致性哈希、分布式文件存储或者增量更新等技术, 实现加速获取代码文件。类似地, 数据获取模块 101 也可以是通过分布式缓存、一致性哈希、分布式文件存储或者增量更新等技术, 从依赖库 202 中获取构建软件所需的依赖文件。

实际应用时, 数据获取模块 1011 也可以是通过其它方式获取代码文件以及依赖文件, 本实施例对此并不进行限定。

数据获取模块 1011 在获取到代码文件后, 可以将该代码文件提供给生成模块 1012, 将依赖文件提供给执行模块 1013。

S203: 生成模块 1012 基于代码文件中的多个目标, 生成多个构建任务, 该多个构建任务中的不同构建任务包括不同的目标。

示例性地, 用户可以通过客户端向实例 101 发送针对软件项目的构建命令, 如 Cmake 命令、AOSP 项目中 Soong 工具的命令等, 从而实例 101 (具体可以是生成模块 1012) 在构建框架中可以生成相应的编译控制文件, 如 cmakelist 文件、ninja 文件等。通常情况下, 编译控制文件中定义有多个目标, 并指示该多个目标的依赖信息 (dependencies), 该依赖信息用于指示要生成目标所需的文件或者其他目标, 即 A 目标的构建成功, 需要 A 目标所依赖的其他目标构建成功。

另外, 由于构建目标通常需要环境变量 (environment variables), 其中, 环境变量, 是指在操作系统中用来指定操作系统运行环境的一些参数, 如临时文件夹位置和系统文件夹位置等。因此, 数据获取模块 1011 还可以获取构建编译控制文件中多个目标分别所需的环境变量。

生成模块 1012 在获取到编译控制文件后, 可以根据编译控制文件生成多个构建任务, 该多个构建任务中的不同构建任务包括不同的目标。

可以理解的是, 如果由单机依次构建该编译控制文件中的各个目标, 则这种串行的构建过程所需耗时较高, 影响软件构建效率。因此, 生成模块 1012 可以是根据编译控制文件中定义的目标以及目标的依赖关系, 生成多个构建任务, 以便后续通过并行执行多个构建任务的方式来加快整体的软件构建效率。

在第一种生成构建任务的实施方式中, 生成模块 1012 可以根据编译控制文件, 生成目标依赖信息树, 该目标依赖信息树通过树形结构的方式指示多个目标的依赖关系, 包括目标之间的依赖关系以及目标所依赖的其他文件等。然后, 生成模块 1012 可以根据该目标依赖信息树, 生成多个构建任务。其中, 每个构建任务可以包括一个目标, 用于负责完成一个目标的构建。并且, 构建任务中还指示构建其包括的目标所需依赖的文件或者其他目标。如此, 针对编译控制文件中定义的多个目标, 生成模块 1012 可以生成相应的多个构建任务。

在第二种生成构建任务的实施方式中, 生成模块 1012 可以根据编译控制文件对应的目标依赖信息树, 将编译控制文件中定义的多个目标进行分组, 具体可以是具有直接或者间接依赖关系的目标划入同一分组, 每个分组中至少包括一个目标 (没有依赖关系的可以单独划入一组), 从而生成模块 1012 可以基于多个分组生成多个构建任务, 每个分组用于生成一个构建任务。

其中, 在对多个目标进行分组时, 生成模块 1012 可以是根据多个目标的构建产物所属的目录进行分组, 具体可以是将构建产物位于同一目录下的一个或者多个目标划入一组。实际场景中, 按照构建产物所属的目录对多个目标进行分组以及构建后, 可以提高针对该多个目标的构建效率。

当然, 生成模块 1012 也可以是基于其他实现方式对多个目标进行分组。比如, 生成模块 1012 也可以是将属于同一工程目录的目标划入同一分组, 不同分组中的目标属于不同的工程目录。或者, 当多个构建任务由多个客户端并行执行时, 生成模块 1012 也可以是根据客户端的数量进行分组, 如客户端的数量为 3 时, 生成模块 1012 可以将多个目标划分为 3 组, 以此生成 3 个构建任务等。

值得注意的是, 上述生成多个构建任务的实现方式仅作为一些示例性说明, 实际应用时, 还可以通过对上述多种实现方式进行组合, 或者采用其他可适用的方式生成多个构建任务, 本实施例对此并不进行限定。

本实施例中, 用于生成构建任务的多个目标, 可以是编译控制文件中的全部目标。或者, 用于生成构建任务的多个目标, 也可以是编译控制文件中的部分目标。比如, 生成模块 1012, 可以先从编译控制文件中筛选出满足预设条件的多个目标, 并根据该多个目标生成多个构建任务, 而编译控制文件中的其

余目标不满足该预设条件。

示例性地，上述预设条件，例如可以是预设类型或者预设标识。比如，针对在 Cmake 构架框架下生成编译控制文件中，预设类型具体可以是 costumer 类型、external project 类型等，即生成模块 1012 可以从中筛选出属于 costumer 类型或者属于 external project 类型的多个目标。又比如，预设标识，具体可以是文件名称等，从而生成模块 1012 可以从编译控制文件中，筛选出与预先指定的文件名称相匹配的多个目标。

实际应用场景中，在进行软件构建之前，用户 200 可以对上述预设条件进行配置。具体地，实例 101 中的配置模块 105 可以生成配置界面，例如可以是如图 3 所示的配置界面，并将该配置界面输出至客户端，以便在客户端上呈现该配置界面。这样，用户 200 可以在该配置界面上针对预设条件执行相应的配置操作，如可以从呈现的多个候选类型中选择部分类型作为上述预设类型。或者，用户 200 可以在图 3 所示的配置界面上输入指定的文件名称，所输入的文件名称作为上述预设标识，或者从呈现的多种文件名称中选择部分文件名称作为上述预设标识等。如此，配置模块 105 可以根据用户 200 执行的配置操作，确定预设条件，并将所确定的预设条件提供给生成模块 1012，以便生成模块 1012 根据该预设条件从编译控制文件中筛选出多个目标，并生成相应的多个构建任务。

生成模块 1012 在生成多个构建任务后，可以将其提供给执行模块 1013。

S204：执行模块 1013 指示资源池 10 中的多个实例利用依赖文件并行执行多个构建任务，得到多个构建任务分别对应的执行结果。

其中，多个构建任务对应的执行结果，不仅可以包括用于指示构建任务执行成功或者失败的信息，还包括对构建任务中的目标完成构建后所生成的构建产物，该构建产物例如可以是实现一种或者多种功能的一组文件，具体可以是 jar 包、war 包等，或者可以是一组“.o”格式的文件、二进制文件、可执行文件、带有后缀扩展名的文件等，对此并不进行限定。

本实施例中，通过并行执行多个构建任务，来提高软件构建的整体效率。

在一种可能的实施方式中，执行模块 1013 可以根据参与并行执行构建任务的多个实例的数量、预先设置的调度规则或者构建任务的相关信息，将多个构建任务分配至多个实例上，每个实例可以负责执行一个或者多个构建任务。

示例性地，执行模块 1013 在根据参与并行执行构建任务的多个实例的数量分配构建任务时，例如可以是将构建任务尽可能均匀的分配到每个实例上，即每个实例上分配到的构建任务的数量可以尽可能一致。比如，当构建任务的数量为 10，而实例的数量为 5 时，则每个实例可以被均分到 2 个构建任务进行执行。

当执行模块 1013 根据预先设置的调度规则分配构建任务时，该调度规则例如可以是随机分配规则（即针对每个构建任务，将其随机分配到某个实例上）、负载均衡规则（即多个实例分配到构建任务后所产生的负载尽可能均衡）等，或者可以是其他可适用的规则。

当执行模块 1013 根据构建任务的相关信息分配构建任务时，该相关信息例如可以是不同构建任务中的目标之间的依赖关系。比如，假设多个构建任务基于目标依赖信息树生成的，则部分构建任务中的目标，可能为目标依赖信息树上作为叶子节点的目标，而其余构建任务中的目标作为该叶子节点的上层节点。此时，执行模块 1013 可以在分配构建任务时，可以先将作为叶子节点的目标所在的构建任务分配至多个实例上进行并行执行。当这部分构建任务执行完成后，多个实例可以向执行模块 1013 反馈构建任务执行成功的反馈，从而执行模块 1013 再根据目标依赖信息树上的节点之间的依赖关系，将剩余的构建任务分配至多个实例上继续进行并行执行，以便多个实例能够利用已执行的构建任务所生成的执行结果继续执行剩余的构建任务。如此，执行模块 1013 以及多个实例之间可以通过循环执行上述分配以及执行构建任务的过程，完成所有构建任务的执行，得到所有构建任务分别对应的执行结果。

上述分配构建任务的实现方式仅作为一些示例性说明，在其他实施例中，也可以是通过对上述多种实现方式进行组合，或者采用其他可适用的方式分配构建任务，对此不做限定。

并且，执行模块 1013 在向多个实例分配构建任务时，还可以将执行构建任务所需的依赖文件一并发送给各个实例，以便各个实例能够利用依赖文件执行其分配到的构建任务。或者，资源池 10 中的依赖文件可以被多个实例所共享，从而各个实例可以基于共享的依赖文件执行构建任务。

值得注意的是，每个实例可以同时分配到一个或者多个构建任务。其中，当每个实例分配到多个构建任务时，该实例可以利用依赖文件依次执行各个构建任务。其中，实例在执行构建任务时，具体是利用依赖文件对该构建任务中包括的一个或者多个目标执行构建过程，其针对目标所执行的具体构建操作，在实际应用场景中存在相关应用，在此不做展开详述。或者，各个实例可以并行执行分配到的多个构建任务。比如，每个实例内部可以创建有多个线程，每个线程可以负责处理一个构建任务，从而各个实例在分配到多个构建任务后，可以利用多个线程并行执行其所分配到的多个构建任务。实际应用时，可以基于不同构建任务中的目标之间的依赖性，确定在实例上串行/并行执行构建任务。

本实施例中，当多个构建任务中的目标为编译控制文件中定义的全部目标时，执行模块 1013 可以将所有构建任务均分配至多个实例上进行执行。

或者，当多个构建任务中的目标为编译控制文件中定义的部分目标时，执行模块 1013 可以将部分目标作为构建任务并将其分配至多个实例上执行，而编译控制文件中的剩余目标可以由执行模块 1013 执行。此时，实例 101 也参与执行构建任务。实际应用场景中，由于实例 101 与其余实例之间的算力差异，对于编译控制文件中的不同目标，其在实例 101 以及其它实例上完成构建的效率可能存在差异，即实例 101 执行部分目标的效率可能高于其余实例执行该部分目标的效率。因此，执行模块 1013 可以根据执行目标所需的算力要求，将编译控制文件中部分目标作为构建任务分配至其它多个实例上完成构建，而其余目标可以由实例 101 完成构建，以此可以进一步提高软件的构建效率。

示例性地，当编译控制文件具体为 `cmakelist` 文件时，生成模块 1012 所生成的多个构建任务中的目标，为 `cmakelist` 文件中定义的部分目标，如可以是属于指定类型（如 `ExternalProject` 类型、`CustomTarget` 类型等）或者是具有指定标识的部分目标，从而由资源池 10 中除实例 101 之外的其余多个实例并行执行多个构建任务，而由实例 101 执行编译控制文件中的剩余目标。并且，在执行模块 1013 分配构建任务的过程中，每个构建任务可以包括一个目标，而执行模块 1013 可以按照目标之间的依赖关系，向多个实例逐个分配构建任务，以便该多个实例在构建作为叶子节点的目标后再构建作为该叶子节点的上层节点的目标。

或者，当编译控制文件具体为 `ninja` 文件时，多个构建任务中的目标为 `ninja` 文件中定义的部分目标，则，生成模块 1012 在生成多个构建任务时，还可以生成新的第一 `ninja` 文件以及第二 `ninja` 文件，并将第一 `ninja` 文件与多个构建任务一并发送给多个实例。并且，每个实例所分配到的构建任务可以包括一组具有依赖关系的目标。这样，执行模块 1013 可以为各个实例一次性分配所需构建的多个目标，无需根据执行结果进行逐个分配。各个实例根据该第一 `ninja` 文件执行构建任务，该第一 `ninja` 文件中定义多个构建任务中的目标以及目标的依赖信息。同时，执行模块 1013 可以根据第二 `ninja` 文件执行编译控制文件中的剩余目标，该第二 `ninja` 文件中定义原 `ninja` 中文件的剩余目标以及该目标的依赖信息。

每个实例在执行各个构建任务并生成相应的执行结果后，可以将该执行结果同步至其他实例，以便其他实例可以基于同步得到的执行结果，执行该其他实例所分配到的构建任务。如此，可以避免单个实例节点出现依赖瓶颈时，影响其余实例针对其所分配到的目标的构建效率，从而保障针对软件的整体构建效率。而当实例 101 也执行部分目标的构建操作时，其余实例还可以将执行构建任务所生成的执行结果也同步至实例 101。

执行模块 1013 在生成多个构建任务分别对应的执行结果后，可以将其提供给构建模块 1014。

S205：构建模块 1014 根据多个构建任务分别对应的执行结果，构建得到软件。

其中，所构建得到的软件，也即用户所要开发的软件。

具体实现时，当多个构建任务中的目标，为编译控制文件中定义的部分目标时，构建模块 1014 可以汇总该多个构建任务分别对应的执行结果，并根据汇总的执行结果，继续完成编译控制文件中剩余目标的构建，以此完成编译控制文件中所有目标的构建，从而构建得到软件。而当多个构建任务中的目标为编译控制文件中定义的全部目标时，构建模块 1014 可以通过汇总各个构建任务分别对应的执行结果，得到针对软件的构建结果。

本实施例中，由于在构建软件的过程中，实例 101 对包括不同目标的多个构建任务并行执行构建过程，这相比于在单机上逐个构建目标的方式，可以有效提升针对该多个目标的构建效率，以此可以有效提升构建软件的整体效率。

需要说明的是，图2中所示的步骤执行顺序仅作为一种实现示例，并不用于进行限定。比如，在其它可能的实施方式中，实例101也可以先执行步骤S202，再执行步骤S201，或者同时执行步骤S201以及步骤S202等。

另外，本实施例中，对于实例101内的各个模块的划分以及功能描述仅作为一种示例。比如，在其它实施例中，数据获取模块1011可以用于执行上述构建软件的方法中的任意步骤，类似的，生成模块1012、执行模块1013、构建模块1014均可以用于执行上述构建软件的方法中的任意步骤，并且，数据获取模块1011、生成模块1012、执行模块1013、构建模块1014负责实现的步骤可根据需要指定，通过数据获取模块1011、生成模块1012、执行模块1013、构建模块1014分别实现构建软件的方法中不同的步骤来实现实例101的全部功能。

上述图1以及图2所示实施例中，针对构建软件的过程中所涉及到的实例101（包括上述数据获取模块1011、生成模块1012、执行模块1013、构建模块1014）可以是配置于计算设备或者计算设备集群上的软件，并且，通过在计算设备或者计算设备集群上运行该软件，可以使得计算设备或者计算设备集群实现上述实例101所具有的功能。下面，基于硬件设备实现的角度，对构建软件的过程中所涉及到的实例101进行详细介绍。

图4示出了一种计算设备的结构示意图，上述实例101可以部署在该计算设备上，该计算设备可以是云环境中的计算设备（如服务器），或边缘环境中的计算设备，或终端设备等具体可以用于实现上述图1所示实施例中数据获取模块1011、生成模块1012、执行模块1013、构建模块1014的功能。

如图4所示，计算设备400包括处理器410、存储器420、通信接口430和总线440。处理器410、存储器420和通信接口430之间通过总线440通信。总线440可以是外设部件互连标准（peripheral component interconnect, PCI）总线或扩展工业标准结构（extended industry standard architecture, EISA）总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示，图4中仅用一条粗线表示，但并不表示仅有一根总线或一种类型的总线。通信接口430用于与外部通信，例如从代码仓中获取代码文件，从依赖库中获取依赖文件等。

其中，处理器410可以为中央处理器（central processing unit, CPU）、专用集成电路（application specific integrated circuit, ASIC）、图形处理器（graphics processing unit, GPU）或者一个或多个集成电路。处理器410还可以是一种集成电路芯片，具有信号的处理能力。在实现过程中，实例101中各个模块的功能可以通过处理器410中的硬件的集成逻辑电路或者软件形式的指令完成。处理器410还可以是通用处理器、数据信号处理器（digital signal process, DSP）、现场可编程逻辑门阵列（field programmable gate array, FPGA）或者其他可编程逻辑器件，分立门或者晶体管逻辑器件，分立硬件组件，可以实现或者执行本申请实施例中公开的方法、步骤及逻辑框图。其中，通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等，结合本申请实施例所公开的方法可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器、闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器420，处理器410读取存储器420中的信息，结合其硬件完成实例101中的部分或全部功能。

存储器420可以包括易失性存储器（volatile memory），例如随机存取存储器（random access memory, RAM）。存储器420还可以包括非易失性存储器（non-volatile memory），例如只读存储器（read-only memory, ROM），快闪存储器，HDD或SSD。

存储器420中存储有可执行代码，处理器410执行该可执行代码以执行前述实例101所执行的方法。

具体地，在实现图2所示实施例的情况下，且图2所示实施例中所述的数据获取模块1011、生成模块1012、执行模块1013、构建模块1014为通过软件实现的情况下，执行图2中的数据获取模块1011、生成模块1012、执行模块1013、构建模块1014的功能所需的软件或程序代码存储在存储器420中，数据获取模块1011与其它设备的交互通过通信接口430实现，处理器用于执行存储器420中的指令，实现实例101所执行的方法。

图5示出的一种计算设备集群的结构示意图。其中，图5所示的计算设备集群50包括多个计算设备，上述实例101可以分布式地部署在该计算设备集群50中的多个计算设备上。如图5所示，计算设

备集群 50 包括多个计算设备 500，每个计算设备 500 包括存储器 520、处理器 510、通信接口 530 以及总线 540，其中，存储器 520、处理器 510、通信接口 530 通过总线 540 实现彼此之间的通信连接。

处理器 510 可以采用 CPU、GPU、ASIC 或者一个或多个集成电路。处理器 510 还可以是一种集成电路芯片，具有信号的处理能力。在实现过程中，实例 101 的部分功能可用通过处理器 510 中的硬件的集成逻辑电路或者软件形式的指令完成。处理器 510 还可以是 DSP、FPGA、通用处理器、其他可编程逻辑器件，分立门或者晶体管逻辑器件、分立硬件组件，可以实现或者执行本申请实施例中公开的部分方法、步骤及逻辑框图。其中，通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等，结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器、闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器 520，在每个计算设备 500 中，处理器 510 读取存储器 520 中的信息，结合其硬件可以完成实例 101 的部分功能。

存储器 520 可以包括 ROM、RAM、静态存储设备、动态存储设备、硬盘（例如 SSD、HDD）等。存储器 520 可以存储程序代码，例如，用于实现数据获取模块 1011 的部分或者全部程序代码、用于实现生成模块 1012 的部分或者全部程序代码、用于实现执行模块 1013 的部分或者全部程序代码、用于实现构建模块 1014 的部分或者全部程序代码等。针对每个计算设备 500，当存储器 520 中存储的程序代码被处理器 510 执行时，处理器 510 基于通信接口 530 执行实例 101 所执行的部分方法，如其中一部分计算设备 500 可以用于执行上述数据获取模块 1011、生成模块 1012 所执行的方法，另一部分计算设备 500 用于执行上述执行模块 1013、构建模块 1014 所执行的方法。存储器 520 还可以存储数据，例如：处理器 510 在执行过程中产生的中间数据或结果数据，例如，上述代码文件、依赖文件、所构建出的软件等。

每个计算设备 500 中的通信接口 530 用于与外部通信，例如与其它计算设备 500 进行交互等。

总线 540 可以是外设部件互连标准总线或扩展工业标准结构总线等。为便于表示，图 5 中每个计算设备 500 内的总线 540 仅用一条粗线表示，但并不表示仅有一根总线或一种类型的总线。

上述多个计算设备 500 之间通过通信网络建立通信通路，以实现实例 101 的功能。任一计算设备可以是云环境中的计算设备（例如，服务器），或边缘环境中的计算设备，或终端设备。

此外，本申请实施例还提供了一种计算机可读存储介质，该计算机可读存储介质中存储有指令，当其在一个或者多个计算设备上运行时，使得该一个或者多个计算设备执行上述实施例中实例 101 包括的各个模块所执行的方法。

此外，本申请实施例还提供了一种计算机程序产品，所述计算机程序产品被一个或者多个计算设备执行时，所述一个或者多个计算设备执行前述构建软件的方法中的任一方法。该计算机程序产品可以为一个软件安装包，在需要使用前述构建软件的方法的任一方法的情况下，可以下载该计算机程序产品并在计算机上执行该计算机程序产品。

另外需说明的是，以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。另外，本申请提供的装置实施例附图中，模块之间的连接关系表示它们之间具有通信连接，具体可以实现为一条或多条通信总线或信号线。

本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的术语在适当情况下可以互换，这仅仅是描述本申请的实施例中对相同属性的对象在描述时所采用的区分方式。

通过以上的实施方式的描述，所属领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件的方式来实现，当然也可以通过专用硬件包括专用集成电路、专用 CPU、专用存储器、专用元器件等来实现。一般情况下，凡由计算机程序完成的功能都可以很容易地用相应的硬件来实现，而且，用来实现同一功能的具体硬件结构也可以是多种多样的，例如模拟电路、数字电路或专用电路等。但是，对本申请而言更多情况下软件程序实现是更佳的实施方式。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在可读取的

存储介质中，如计算机的软盘、U 盘、移动硬盘、ROM、RAM、磁碟或者光盘等，包括若干指令用以使得一台计算机设备（可以是个人计算机，训练设备，或者网络设备）执行本申请各个实施例所述的方法。

在上述实施例中，可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。

所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一计算机可读存储介质传输，例如，所述计算机指令可以从一个网站站点、计算机、训练设备或数据中心通过有线（例如同轴电缆、光纤、数字用户线（DSL））或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、训练设备或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存储的任何可用介质或者是包含一个或多个可用介质集成的训练设备、数据中心等数据存储设备。所述可用介质可以是磁性介质，（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质（例如固态硬盘（Solid State Disk，SSD））等。

权 利 要 求 书

1. 一种构建软件的方法，其特征在于，所述方法应用于资源池中的第一实例，所述方法包括：

所述第一实例从代码仓中获取待构建的软件的代码文件；

所述第一实例从依赖库中获取构建所述软件所需的依赖文件；

5 所述第一实例基于所述代码文件中的多个目标，生成多个构建任务，所述多个构建任务中的不同构建任务包括不同的目标；

所述第一实例指示所述资源池中的多个实例利用所述依赖文件并行执行所述多个构建任务，得到所述多个构建任务分别对应的执行结果；

所述第一实例根据所述多个构建任务分别对应的执行结果，构建得到软件。

10 2. 根据权利要求1所述的方法，其特征在于，所述第一实例从代码仓中获取待构建的软件的代码文件，包括：

所述第一实例对所述代码仓中的所述软件的代码文件进行分布式缓存。

3. 根据权利要求1或2所述的方法，其特征在于，所述第一实例基于所述代码文件中的多个目标，生成多个构建任务，包括：

15 所述第一实例确定所述代码文件中的编译控制文件，所述编译控制文件包括多个目标以及所述多个目标的依赖信息；

所述第一实例根据所述编译控制文件，为所述软件生成所述多个构建任务。

4. 根据权利要求3所述的方法，其特征在于，所述多个目标为所述编译控制文件中定义的全部目标；

20 或者，所述多个目标为所述编译控制文件中满足预设条件的目标，所述预设条件包括预设类型或者预设标识。

5. 根据权利要求4所述的方法，其特征在于，所述方法还包括：

所述第一实例输出配置界面；

所述第一实例响应于用户在所述配置界面上针对预设条件的配置操作，确定所述预设条件。

25 6. 根据权利要求3至5任一项所述的方法，其特征在于，所述第一实例指示所述资源池中的多个实例利用所述依赖文件并行执行所述多个构建任务，得到所述多个构建任务分别对应的执行结果，包括：

所述第一实例按照预设规则，将所述多个构建任务分发给所述多个实例，所述多个构建任务被所述多个实例并行执行，所述多个实例中的不同实例执行不同的构建任务；

所述第一实例获取所述多个实例反馈的所述多个构建任务分别对应的执行结果。

30 7. 根据权利要求3至6任一项所述的方法，其特征在于，所述第一实例基于所述代码文件中的多个目标，生成多个构建任务，包括：

所述第一实例根据所述多个实例的数量或所述多个目标的构建产物所属的目录，对所述编译控制文件中的多个目标进行分组，得到所述多个构建任务，每个构建任务包括的目标为所述多个目标中的一组目标。

35 8. 根据权利要求6或7所述的方法，其特征在于，所述编译控制文件中还包括除所述多个目标之外的其他目标；

所述第一实例根据所述多个构建任务分别对应的执行结果，构建得到软件，包括：

所述第一实例构建所述编译控制文件中的其他目标，得到本地执行结果；

所述第一实例根据所述本地执行结果以及所述多个构建任务分别对应的执行结果，构建得到所述软件。

40 9. 一种构建软件的方法，其特征在于，所述方法包括：

获取待构建的软件的代码文件中的编译控制文件以及构建所述软件所需的依赖文件，所述编译控制文件包括多个目标以及所述多个目标的依赖信息；

根据所述编译控制文件，为所述软件生成多个构建任务，所述多个构建任务中的不同构建任务包括不同的目标；

45 并行执行所述多个构建任务，得到所述多个构建任务分别对应的执行结果；

根据所述多个构建任务分别对应的执行结果，构建得到软件。

10. 一种软件构建装置，其特征在于，所述软件构建装置应用于资源池中的第一实例，所述软件构建装置包括：

数据获取模块，用于从代码仓中获取待构建的软件的代码文件，并从依赖库中获取构建所述软件所需的依赖文件；

生成模块，用于基于所述代码文件中的多个目标，生成多个构建任务，所述多个构建任务中的不同构建任务包括不同的目标；

执行模块，用于指示所述资源池中的多个实例利用所述依赖文件并行执行所述多个构建任务，得到所述多个构建任务分别对应的执行结果；

构建模块，用于根据所述多个构建任务分别对应的执行结果，构建得到软件。

11. 根据权利要求 10 所述的软件构建装置，其特征在于，所述数据获取模块，用于对所述代码仓中的所述软件的代码文件进行分布式缓存。

12. 根据权利要求 10 或 11 所述的软件构建装置，其特征在于，所述生成模块，用于：

确定所述代码文件中的编译控制文件，所述编译控制文件包括多个目标以及所述多个目标的依赖信息；

根据所述编译控制文件，为所述软件生成所述多个构建任务。

13. 根据权利要求 12 所述的软件构建装置，其特征在于，所述多个目标为所述编译控制文件中定义的全部目标；

或者，所述多个目标为所述编译控制文件中满足预设条件的目标，所述预设条件包括预设类型或者预设标识。

14. 根据权利要求 13 所述的软件构建装置，其特征在于，所述软件构建装置还包括：

配置模块，用于输出配置界面；以及响应于用户在所述配置界面上针对预设条件的配置操作，确定所述预设条件。

15. 根据权利要求 12 至 14 任一项所述的软件构建装置，其特征在于，所述执行模块，用于：

按照预设规则，将所述多个构建任务分发给所述多个实例，所述多个构建任务被所述多个实例并行执行，所述多个实例中的不同实例执行不同的构建任务；

获取所述多个实例反馈的所述多个构建任务分别对应的执行结果。

16. 根据权利要求 12 至 15 任一项所述的软件构建装置，其特征在于，所述生成模块，用于根据所述多个实例的数量或所述多个目标的构建产物所属的目录，对所述编译控制文件中的多个目标进行分组，得到所述多个构建任务，每个构建任务包括的目标为所述多个目标中的一组目标。

17. 根据权利要求 15 或 16 所述的软件构建装置，其特征在于，所述编译控制文件中还包括除所述多个目标之外的其他目标；

所述构建模块，用于：

构建所述编译控制文件中的其他目标，得到本地执行结果；

根据所述本地执行结果以及所述多个构建任务分别对应的执行结果，构建得到所述软件。

18. 一种软件构建装置，其特征在于，所述软件构建装置，包括：

数据获取模块，用于获取待构建的软件的代码文件中的编译控制文件以及构建所述软件所需的依赖文件，所述编译控制文件包括多个目标以及所述多个目标的依赖信息；

生成模块，用于根据所述编译控制文件，为所述软件生成多个构建任务，所述多个构建任务中的不同构建任务包括不同的目标；

执行模块，用于并行执行所述多个构建任务，得到所述多个构建任务分别对应的执行结果；

构建模块，用于根据所述多个构建任务分别对应的执行结果，构建得到软件。

19. 一种计算设备集群，其特征在于，包括至少一个计算设备，每个计算设备包括处理器和存储器；

所述处理器用于执行所述存储器中存储的指令，以使得所述计算设备集群执行权利要求 1 至 9 中任一项所述的方法。

20. 一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中存储有指令，当其在至少

一个计算设备上运行时，使得所述至少一个计算设备执行如权利要求 1 至 9 任一项所述的方法。

21. 一种包含指令的计算机程序产品，其特征在于，当其在至少一个计算设备上运行时，使得所述至少一个计算设备执行如权利要求 1 至 9 中任一项所述的方法。

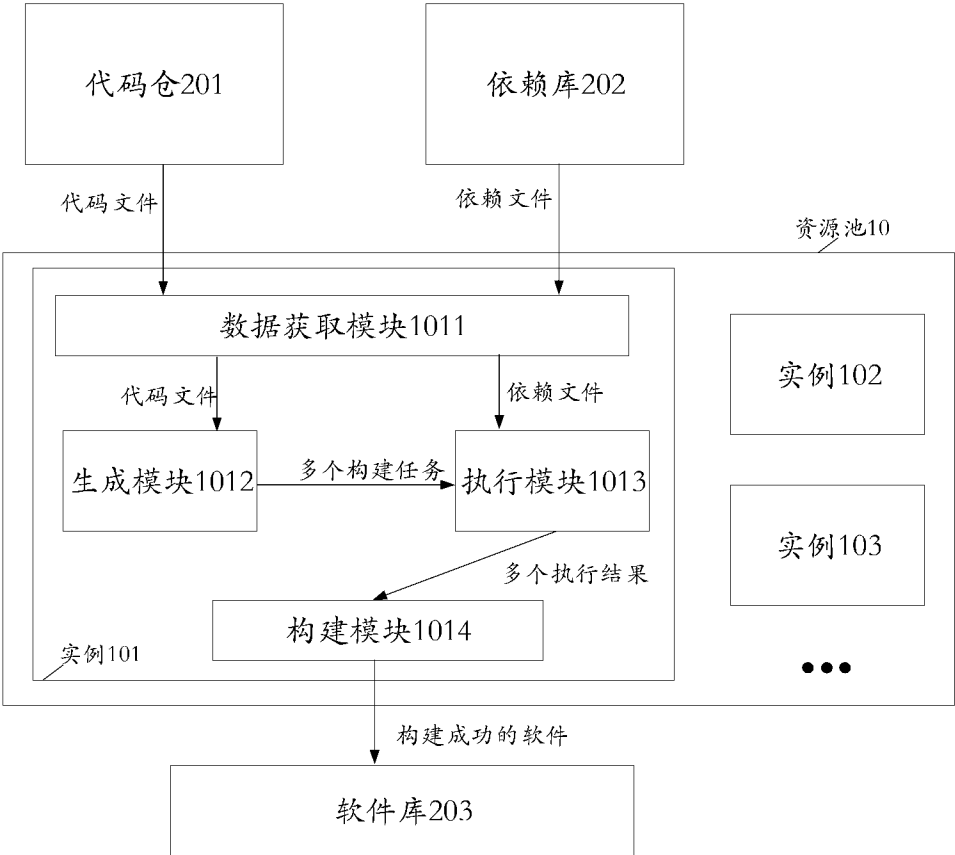


图 1

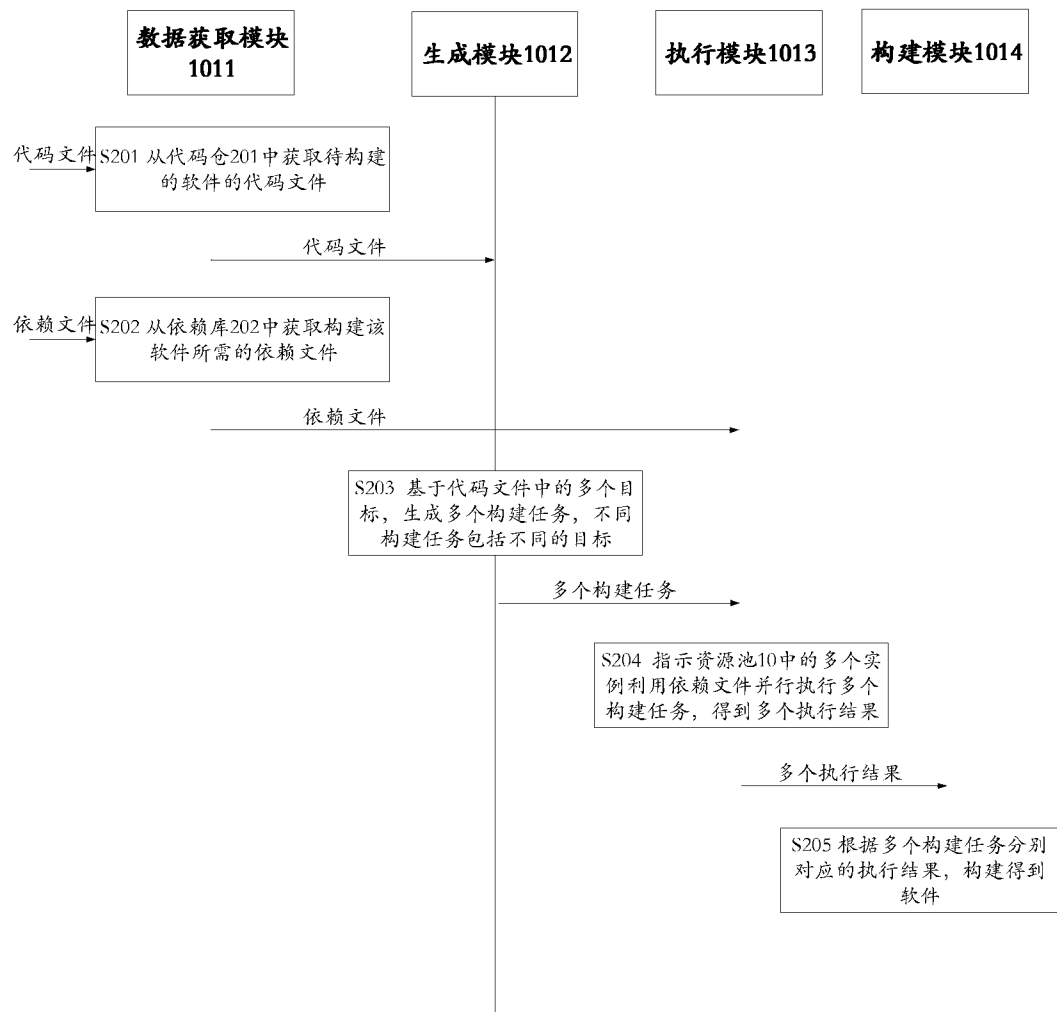


图 2

配置界面

请选择并行构建的目标类型！

候选类型1

☒

候选类型2

☐

候选类型3

☒

候选类型4

☐

请输入并行构建的目标名称！

请选择并行构建的目标名称！

文件名称1

☒

文件名称2

☒

文件名称3

☐

文件名称4

☐

确定

取消

图 3

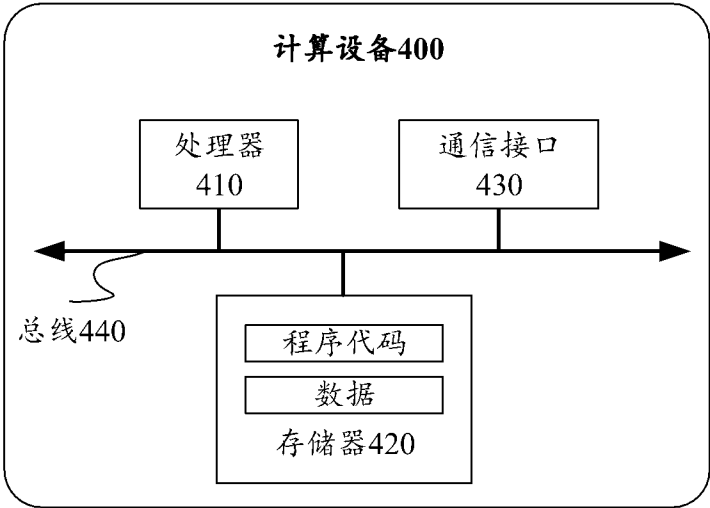


图 4

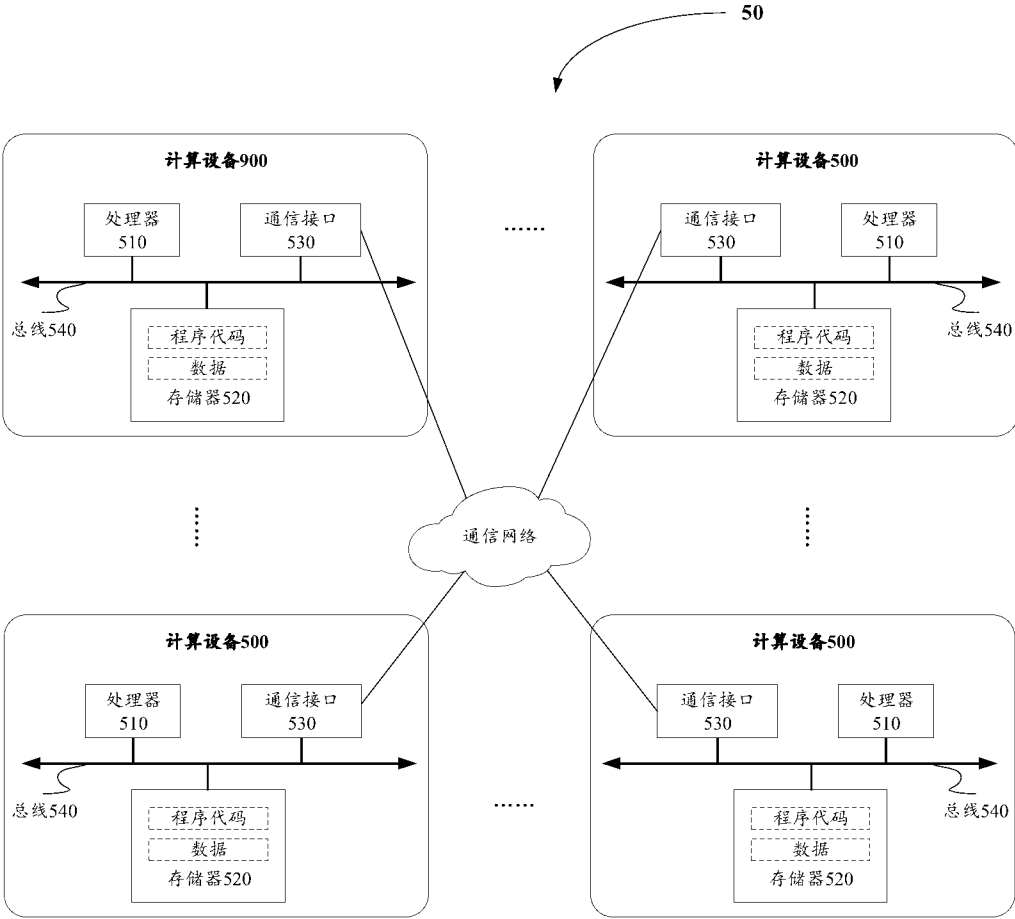


图 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2023/120630

A. CLASSIFICATION OF SUBJECT MATTER		
G06F8/20(2018.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
CNTXT, ENTXT, ENTXT, DWPI, CNKI, IEEE, ACM, 百度, BAIDU: 构建, 编译, 并行, 分布式, 控制, 目标, 配置, 任务, 软件, 应用, build, compile, parallel, distributed, control, target, configuration, task, software, application, bazel, ninja, cmake		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	放浪者 (Bohemian). "Bazel入门 (Non-official translation: Bazel Entry)" https://zhuanlan.zhihu.com/p/262497747 , 09 October 2020 (2020-10-09), pages 1-8	1-21
Y	US 2007234320 A1 (EBAY INC.) 04 October 2007 (2007-10-04) description, paragraphs 24-46, and figures 1-7	1-8, 10-17, 19-21
Y	生栋说 (Shengdong say). "Google分布式构建软件之二 (Non-official translation: Google Distributed Build Software II)" https://www.cnblogs.com/Jack47/p/build-in-the-cloud_how-the-build-system-works.html , 01 April 2015 (2015-04-01), pages 1-3	3-9, 12-21
A	CN 111338686 A (PING AN LIFE INSURANCE COMPANY OF CHINA, LTD.) 26 June 2020 (2020-06-26) entire document	1-21
A	CN 105975276 A (SHANGHAI CTRIP COMMERCE CO., LTD.) 28 September 2016 (2016-09-28) entire document	1-21
☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "D" document cited by the applicant in the international application "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
23 October 2023		01 December 2023
Name and mailing address of the ISA/CN		Authorized officer
China National Intellectual Property Administration (ISA/CN) China No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088		Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2023/120630

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 20110116772 A (NEMUSTECH) 26 October 2011 (2011-10-26) entire document	1-21

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2023/120630

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2007234320	A1	04 October 2007	US	2017097821	A1	06 April 2017
				US	10048961	B2	14 August 2018
				US	2015169322	A1	18 June 2015
				US	9529589	B2	27 December 2016
				US	2018365005	A1	20 December 2018
				US	10620943	B2	14 April 2020
				US	2020233660	A1	23 July 2020
				US	11106459	B2	31 August 2021
				US	2009070750	A9	12 March 2009
				US	8458661	B2	04 June 2013
				US	2013268915	A1	10 October 2013
				US	8972937	B2	03 March 2015
CN	111338686	A	26 June 2020	None			
CN	105975276	A	28 September 2016	CN	105975276	B	09 April 2019
KR	20110116772	A	26 October 2011	None			

A. 主题的分类 G06F8/20(2018.01)i 按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类		
B. 检索领域 检索的最低限度文献(标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用)) CNTXT,ENTXTC,ENTXT,DWPI,CNKI,IEEE,ACM,百度:构建,编译,并行,分布式,控制,目标,配置,任务,软件,应用,build, compile,parallel,distributed,control,target,configuration,task,software,application,bazel,ninja,cmakelist		
C. 相关文件		
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
Y	放浪者. "Bazel入门" https://zhuanlan.zhihu.com/p/262497747 , 2020年10月9日 (2020 - 10 - 09), 第1-8页	1-21
Y	US 2007234320 A1 (EBAY INC) 2007年10月4日 (2007 - 10 - 04) 说明书第24-46段, 图1-7	1-8,10-17,19-21
Y	生栋说. "Google分布式构建软件之二" https://www.cnblogs.com/Jack47/p/build-in-the-cloud_how-the-build-system-wor-ks.html , 2015年4月1日 (2015 - 04 - 01), 第1-3页	3-9,12-21
A	CN 111338686 A (中国平安人寿保险股份有限公司) 2020年6月26日 (2020 - 06 - 26) 全文	1-21
A	CN 105975276 A (上海携程商务有限公司) 2016年9月28日 (2016 - 09 - 28) 全文	1-21
A	KR 20110116772 A (NEMUSTECH) 2011年10月26日 (2011 - 10 - 26) 全文	1-21
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。		
* 引用文件的具体类型: "A" 认为不特别相关的表示了现有技术一般状态的文件 "D" 申请人在国际申请中引证的文件 "E" 在国际申请日的当天或之后公布的在先申请或专利 "L" 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的) "O" 涉及口头公开、使用、展览或其他方式公开的文件 "P" 公布日先于国际申请日但迟于所要求的优先权日的文件 "T" 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 "X" 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 "Y" 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 "&" 同族专利的文件		
国际检索实际完成的日期 2023年10月23日		国际检索报告邮寄日期 2023年12月1日
ISA/CN的名称和邮寄地址 中国国家知识产权局 中国北京市海淀区蓟门桥西土城路6号 100088		授权官员 胡赢 电话号码 (+86) 027-59371827

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2023/120630

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
US	2007234320	A1	2007年10月4日	US	2017097821	A1	2017年4月6日
				US	10048961	B2	2018年8月14日
				US	2015169322	A1	2015年6月18日
				US	9529589	B2	2016年12月27日
				US	2018365005	A1	2018年12月20日
				US	10620943	B2	2020年4月14日
				US	2020233660	A1	2020年7月23日
				US	11106459	B2	2021年8月31日
				US	2009070750	A9	2009年3月12日
				US	8458661	B2	2013年6月4日
				US	2013268915	A1	2013年10月10日
				US	8972937	B2	2015年3月3日
CN	111338686	A	2020年6月26日	无			
CN	105975276	A	2016年9月28日	CN	105975276	B	2019年4月9日
KR	20110116772	A	2011年10月26日	无			