

应用部署方法、装置及设备

本申请要求于 2023 年 12 月 27 日提交国家知识产权局、申请号为 202311823774.X、申请名称为“应用编排方法、装置、计算设备集群及存储介质”的中国专利申请的优先权，以及 2024 年 04 月 16 日提交国家知识产权局、申请号为 202410458081.3、申请名称为“应用部署方法、装置及设备”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

本申请实施例涉及 IT 技术领域，尤其涉及一种应用部署方法、装置及设备。

背景技术

应用编排服务是一种通过图形化编排的方式对应用进行开发的技术。用户在使用应用编排服务时，可以通过简单的拖拉拽等操作，对多个组件进行编排，并对各个组件进行参数配置，以完成应用的开发。

在应用开发完成后，服务器可以基于应用中的每个组件部署该应用。服务器可以创建每个组件对应的容器，由该容器运行对应组件。但是，由于容器在启动时需要占用一定数量的资源，导致服务器在启动容器时便需要为每个容器均分配资源。如此，会导致应用占用大量的资源，降低了资源的利用率。

发明内容

本申请提供一种应用部署方法、装置及设备，能够减少容器的部署数量，从而减少资源的占用。

第一方面，本申请提供一种应用部署方法，该方法中，可以提供组件编排界面，组件编排界面包括：组件选择区和组件编排区，组件选择区包括：同容器组件和多个预设组件，同容器组件用于指示与其自身关联的多个预设组件运行在同一容器中，组件编排区用于为用户提供编排组件的操作区，多个预设组件包括业务组件和/或逻辑组件。之后，响应于用户在组件编排区的操作，可以在组件编排区中创建第一同容器组件，并确定与第一同容器组件关联的多个第一预设组件，多个预设组件包含多个第一预设组件。然后，响应于接收到的应用部署请求，部署包含多个第一预设组件的目标应用，其中，多个第一预设组件运行在第一同容器组件对应的第一容器中。

基于上述技术方案，可以通过第一同容器组件对多个第一预设组件进行编排，并且第一同容器组件和多个第一预设组件均可以运行在同一容器中。这样一来，在部署应用时，服务器可以通过一个容器加载多个组件，减少了容器的部署数量，从而减少容器占用的资源量。并且，由于多个第一预设组件均对应第一容器，则说明一个容器可以运行多个组件，使得多个组件可以在容器中通过内存进行访问，从而提高组件间的访问效率，进而提升应用的流畅度。

结合第一方面，在一种可能的设计方式中，响应于接收到的应用发布请求，根据组件编排区中的全部组件，生成目标应用，全部组件包括同容器组件和多个第一预设组件。

结合第一方面，在另一种可能的设计方式中，组件编排界面还包括：组件配置区。从第一同容器组件对应的组件配置区中获取目标资源量的大小，目标资源量为第一同容器组件对应的第一容器所需的资源量。为第一容器分配目标资源量。

可以理解的是，由于第一同容器组件中包括多个第一预设组件，在创建第一容器时，可以基于第一同容器组件中的部署参数，为第一容器分配目标资源量，以保障第一容器能够运行多个第一预设组件。并且，由于多个第一预设组件运行在一个第一容器中，无需为每个组件的容器单独分配启动时的资源量，从而减少了容器占用的资源数量。

结合第一方面，在另一种可能的设计方式中，解压每个第一预设组件，得到每个第一预设组件对应的解压文件。之后，可以将多个第一预设组件对应的全部解压文件进行压缩，得到融合组件，并根据融合组件，生成目标应用。

可以理解的是，通过将多个第一预设组件进行融合，得到融合组件，使得部署第一容器时，只需要加载融合组件，便可以使得第一容器运行多个第一预设组件，以减少部署的容器数量。

结合第一方面，在另一种可能的设计方式中，可以将调用的第一预设组件的接口均更新为调

用融合组件的接口。

这样一来，其他组件调用第一预设组件时，可以通过调用融合组件以实现第一预设组件的功能，并完成参数的传递。

结合第一方面，在另一种可能的设计方式中，可以根据组件编排区中的全部组件和全部组件中各个组件之间的连接关系，生成目标应用。

结合第一方面，在另一种可能的设计方式中，全部组件包括：多个第一预设组件；全部组件中各个组件之间的连接关系包括：第一预设组件之间的连接关系。

结合第一方面，在另一种可能的设计方式中，全部组件还包括：至少一个第二预设组件，第二预设组件为多个预设组件中任一组件，第二预设组件在第一同容器组件外；全部组件中各个组件之间的连接关系还包括：第二预设组件与边缘组件之间的连接关系，边缘组件为多个第一预设组件中与第二预设组件存在连接关系的组件。

也就是说，组件编排区中不仅包括第一预设组件，还包括第二预设组件。

结合第一方面，在另一种可能的设计方式中，至少一个第二预设组件包括：至少一个第三预设组件，边缘组件包括：边缘输入组件，第三预设组件为向边缘输入组件传递参数的组件。可以获取每个第三预设组件的输出参数。之后，可以将每个第三预设组件的输出参数进行聚合，得到第一同容器组件的输入参数集合。然后，可以根据每个第三预设组件与边缘输入组件之间的连接关系，建立输入参数集合中各个参数与边缘输入组件的输入参数之间的映射关系。

可以理解的是，通过建立输入参数集合中各个参数与边缘输入组件的输入参数之间的映射关系，可以保障应用运行时，容器间能够正常传递参数。

结合第一方面，在另一种可能的设计方式中，至少一个第二预设组件还包括：至少一个第四预设组件，边缘组件还包括：边缘输出组件，边缘输出组件为向第四预设组件传递参数的组件。可以获取每个边缘输出组件的输出参数。之后，将每个边缘输出组件的输出参数进行聚合，得到第一同容器组件的输出参数集合。然后，根据每个第四预设组件与边缘输出组件之间的连接关系，建立输出参数集合中各个参数与第四预设组件的输入参数之间的映射关系。

可以理解的是，通过建立输出参数集合中各个参数与第四预设组件的输入参数之间的映射关系，可以保障应用运行时，容器间能够正常传递参数。

结合第一方面，在另一种可能的设计方式中，在运行第一容器的情况下，多个第一预设组件之间通过目标数据结构传递参数，目标数据结构为在内存中使用的数据结构。

这样一来，由于内存中传递参数的速度较快，多个第一预设组件之间通过目标数据结构在内存中传递参数，能够提高数据的传递效率，降低时延，进而提高应用的运行速度。

第二方面，本申请提供一种应用部署装置，包括：处理模块，用于响应于用户在组件编排区的操作，在组件编排区中创建第一同容器组件，并确定与第一同容器组件关联的多个第一预设组件，多个预设组件包含多个第一预设组件。处理模块，用于响应于接收到的应用部署请求，部署包含多个第一预设组件的目标应用，其中，多个第一预设组件运行在第一同容器组件对应的第一容器中。

第三方面，本申请提供一种计算设备集群，该计算设备集群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。至少一个计算设备的处理器用于执行至少一个计算设备的存储器中存储的指令，使得计算设备集群执行如第一方面及其任一种可能的设计方式所述的方法。

第四方面，本申请提供一种计算机可读存储介质，该计算机可读存储介质包括计算机指令，当所述计算机指令在电子设备上运行时，使得所述电子设备执行如第一方面及其任一种可能的设计方式所述的方法。

第五方面，本申请提供一种计算机程序产品，当所述计算机程序产品在电子设备上运行时，使得所述电子设备执行如第一方面及其任一种可能的设计方式所述的方法。

第六方面，本申请提供一种芯片系统，该芯片系统应用于电子设备。该芯片系统包括一个或多个接口电路和一个或多个处理器。该接口电路和处理器通过线路互联。该接口电路用于从电子设备的存储器接收信号，并向处理器发送该信号，该信号包括存储器中存储的计算机指令。当处理器执行所述计算机指令时，电子设备执行如第一方面及其任一种可能的设计方式所述的方法。

可以理解地，上述提供的第二方面所述的应用部署装置，第三方面所述的计算设备集群，第四方面所述的计算机可读存储介质，第五方面所述的计算机程序产品，第六方面所述的芯片系统所能达到的有益效果，可参考如第一方面及其任一种可能的设计方式中的有益效果，此处不再赘述。

5 附图说明

- 图 1 为本申请实施例提供的一种组件编排的实例示意图；
- 图 2 为本申请实施例提供的一种应用部署系统的组成示意图；
- 图 3 为本申请实施例提供的一种应用部署平台的系统架构示意图；
- 图 4 为本申请实施例提供的一种应用部署方法流程图；
- 10 图 5 为本申请实施例提供的另一种组件编排的实例示意图；
- 图 6 为本申请实施例提供的另一种组件编排的实例示意图；
- 图 7 为本申请实施例提供的另一种组件编排的实例示意图；
- 图 8 为本申请实施例提供的一种组件配置的实例示意图；
- 图 9 为本申请实施例提供的另一种应用部署方法流程图；
- 15 图 10 为本申请实施例提供的一种应用部署方法的实例示意图；
- 图 11 为本申请实施例提供的另一种应用部署方法流程图；
- 图 12 为本申请实施例提供的一种应用部署界面的实例示意图；
- 图 13 为本申请实施例提供的一种容器内运行多组件的实例示意图；
- 图 14 为本申请实施例提供的一种应用部署装置的组成示意图；
- 20 图 15 为本申请实施例提供的一种计算设备的结构组成示意图；
- 图 16 为本申请实施例提供的一种计算设备集群的结构组成示意图；
- 图 17 为本申请实施例提供的另一种计算设备集群的结构组成示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，
25 显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例，都属于本申请保护的范围。

本申请中字符“/”，一般表示前后关联对象是一种“或者”的关系。例如，A/B 可以理解为 A 或者 B。

30 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一个或者更多个该特征。在本实施例的描述中，除非另有说明，“多个”的含义是两个或两个以上。

此外，本申请的描述中所提到的术语“包括”和“具有”以及它们的任何变形，意图在于覆盖不排他的包含。例如包含了一系列步骤或模块的过程、方法、系统、产品或设备没有限定于已列出的
35 的步骤或模块，而是可选地还包括其他没有列出的步骤或模块，或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或模块。

另外，在本申请实施例中，“示例性的”、“或者”“例如”等词用于表示作例子、例证或说明。本申请中被描述为“示例性的”或“例如”的任何实施例或设计方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言，使用“示例性的”、“或者”“例如”等词旨在以具体方式呈现概念。
40 念。

为了便于理解本申请的技术方案， 在本申请实施例的深度图像的获取方法进行详细介绍之前，先对本申请实施例中所提到的专业名词进行介绍。

1、应用编排，是一种图形化编排技术，用户可以通过简单的拖拉拽等操作以及参数配置的方式来对应用进行二次开发，并支持对已开发的应用重新进行组合编排，从而提升应用开发效率。并且，在应用编排完成后，用户可以将编排完成的应用以服务的形式发布，并通过应用程序编程接口（application programming interface， API）的形式对外提供服务。
45

2、画布，是应用编排的载体。用户在画布上能够使用拖拉拽的方式，将各种功能组件进行编排，配置组件参数值，以流程的形式将二次开发的应用展示出来。

3、组件(component)，是低代码开发中的基本构建块，可以由一定业务逻辑、以及相关数据组成。组件可以完成特定的功能。例如，视频播放组件可以完成播放视频的功能、视频解码组件可以完成对视频解码的功能。一个应用(application, APP)可以由一个或多个组件组成。组件可以有用户界面，也可以没有用户界面。

例如，视频解码组件可以没有用户界面，即电子设备通过视频解码组件对视频解码的过程可以不展现在用户界面中，用户不感知。单个组件可以独立运行于用户设备中，例如，电视中可以安装有播放视频组件，然后电视通过该播放视频组件播放视频。又例如，组件可以是用户界面(UI)元素，例如按钮、文本框、表格，也可以是业务逻辑的一部分，例如数据处理、流程控制等。

每个组件包括一个组件主体和若干个连接点(输入连接点或输出连接点)。一个组件通过连接点来实现与其他组件传递数据。通常情况下，一个组件包括至少一个输入连接点和/或至少一个输出连接点。又或者，部分组件也可以没有连接点，即这类组件无输入连接点和输出连接点。在本申请实施例中，可以用来组成应用的组件包括至少一个连接点。

其中，连接点是组件的对外的智能化接口，即实现输入或输出功能的代理者，负责与其他连接点进行接口协议检查、协商对接、数据传递等。连接点是组件接收另一个组件输入数据的途径，也是组件向另一个组件输出数据的途径。连接点包括输入连接点或者输出连接点。通常，每个连接点可以支持一种或多种类型的数据实体 Entity。连接点支持的数据实体可以是图像、音频、视频、文字等等。

对本申请实施例中所提到的专业名词进行介绍之后，下面对常规技术进行介绍。

随着工业数字化进程的推进，应用编排的业务场景也逐渐丰富，应用所需要的组件种类与数量也随之增多。在开发应用的过程中，用户可能会需要较多的组件，从而完成一个应用的开发。目前，用户可以通过简单的拖拉拽等操作，对多个组件进行编排组合，并对各个组件进行参数配置，以完成应用的开发。在部署应用时，服务器可以创建每个组件对应的容器，由该容器运行组件对应的服务。

示例性的，如图 1 所示，画布 101 中包括组件 a、组件 b 和组件 c (均为平台组件，即平台提供的组件)，在部署由组件 a、组件 b 和组件 c 组成的应用 a 时，服务器可以分别部署组件 a 对应的容器 A、组件 b 对应的容器 B、组件 c 对应的容器 C，进而启动应用 a 包括的各个组件对应的容器，以便在使用部署的应用 a 时，从容器中调用该应用 a 包括的组件，实现该应用的功能。

但是，由于容器在启动时需要创建一个独立的运行环境，导致容器在启动时便需要占用一定数量的资源。如此，会导致用户未使用应用时，已经占用了服务器的大量资源，降低了资源的利用率。例如，结合图 1 可知，在服务器启动容器 a、容器 b 和容器 c 时，便需要为每个容器均分配一定的资源。并且，在使用部署的应用时，容器间需要通过网络进行数据传输，从而完成容器间的相互协作。但是，容器间通信会受到网络拓扑、网络带宽、网络负载等影响，可能导致容器间数据传输发生延迟。

综上，上述技术方案中，在组件数量较多的情况下，不仅会导致容器数量变多，从而占用服务器的大量资源；而且容器间数据传输时延较高，导致应用发生卡顿，从而影响用户的使用体验。

为此，本申请实施例提供一种应用部署方法。该方法中，服务器可以提供组件编排界面，该组件编排界面包括组件选择区和组件编排区，组件选择区包括：同容器组件和多个预设组件，同容器组件用于指示与其自身关联的多个预设组件运行在同一容器中，第一同容器组件与第一容器相对应。之后，响应于对第一同容器组件的选择操作，电子设备可以在组件编排区显示第一同容器组件。并响应于对多个预设组件中多个第一预设组件的选择操作，在第一同容器组件中显示多个第一预设组件，多个第一预设组件与第一容器相对应。然后，响应于应用发布操作，电子设备向服务器发送应用发布请求。服务器可以从组件编排界面中获取组件编排区中的全部组件，生成

目标应用。

也就是说，可以通过第一同容器组件对多个第一预设组件进行编排，并且第一同容器组件和多个第一预设组件均可以运行在同一容器中。这样一来，在部署应用时，服务器可以通过一个容器加载多个组件，减少了容器的部署数量，从而减少容器占用的资源量。并且，由于多个第一预设组件均对应第一容器，则说明一个容器可以运行多个组件，使得多个组件可以在容器中通过内存进行访问，从而提高组件间的访问效率，进而提升应用的流畅度。

下面对本申请实施例的实施环境进行介绍。

如图 2 所示，为本申请实施例提供的一种应用部署系统，该应用部署系统包括：计算设备集群 201、至少一个电子设备（如电子设备 202）。计算设备集群 201 可以与电子设备 202 进行无线/有线通信。

其中，计算设备集群 201 可以通过一组松散集成的计算机软件或硬件连接起来高度紧密地协作完成计算工作，也可以将计算设备集群 201 看作是一台计算机。计算设备集群 201 中的单个计算机通常称为节点，节点之间可以通过局域网、高速互联、远程直接内存访问（remote direct memory access，RDMA）、分布式共享内存等方式连接。计算设备集群 201 可以包括高可用性集群、负载均衡集群、高性能计算集群、高可用性集群中的一种或多种集群。计算设备集群 201 可以包括：硬件资源（如服务器、存储器、中央处理器（central processing unit，CPU）等）和服务资源（如软件、集成开发环境等）。

本申请实施例中的电子设备可以是平板电脑、手机、手持计算机、笔记本电脑、超级移动个人计算机（ultra-mobile personal computer，UMPC）、上网本，以及蜂窝电话、个人数字助理（personal digital assistant，PDA）、增强现实（augmented reality，AR）、虚拟现实（virtual reality，VR）设备、车载设备等设备，本申请实施例对该电子设备的具体形态不作特殊限制。

在本申请实施例中，计算设备集群 201 可以用于提供应用开发、部署服务，即对不同的组件进行编排组合，以生成新的应用。例如，该计算设备集群 201 中部署有应用部署平台（或者应用开发软件），该应用部署平台用于对组件进行编排组合，生成应用。计算设备集群 201 可以接收电子设备 202 的应用生成指令，该应用生成指令用于指示通过多个组件生成应用，计算设备集群 201 可以基于该应用生成指令生成应用。并且，计算设备集群 201 可以接收电子设备 202 的部署指令，部署应用。

电子设备 202 可以登录应用部署平台，并提供可视化界面。电子设备 202 可以接收用户的应用管理操作，以结合用户操作指示计算设备集群 201 执行管理应用的指令。例如，电子设备 202 可以显示应用部署平台的界面，用户可以在该应用部署平台的界面进行操作，并由电子设备 202 向计算设备集群 201 发送开发指令或者部署指令等，以指示计算设备集群 201 执行相应的任务。

在对本申请实施例的实施环境进行介绍之后，下面对部署在计算设备集群 201 的应用部署平台进行介绍。

如图 3 所示，为本申请实施例提供的一种应用部署平台（也可以称为应用部署平台），该应用部署平台包括：用户界面(user interface,UI)模块、画布管理模块、组件管理模块、运行管理模块。

其中，UI 模块，用于在 UI 上显示画布，该画布包括：组件选择区、组件编排区。组件选择区用于为用户提供待编排的组件，组件选择区可以包括多个预设组件，组件编排区用于显示用户所选择的组件。该 UI 模块，还用于接收用户的操作执行相应的指令。例如，UI 模块可以接收用户对组件选择区中的组件的拖拉拽等操作，在组件编排区中显示用户操作的组件。又例如，UI 模块还可以包括控件（如配置控件），该配置控件用于接收用户对组件的配置操作，对组件进行配置，并在 UI 显示组件的配置信息。又例如，UI 模块可以接收用户的自定义组件操作，生成新的组件。

可选的，该画布还可以包括：组件配置区，组件配置区用于设置组件的参数，如输入参数、输出参数、部署参数等。

画布管理模块可以提供多个接口：如管理接口、监控接口、发布接口、部署接口。其中，管理接口用于实现对画布中组件的增加、删除等。监控接口用于实现对画布中组件的变化（如数量

和关系的变化)。发布接口用于实现将画布中的组件发布为应用。部署接口用于部署画布中的组件。

5 组件管理模块用于对应用部署平台中的组件进行管理。例如, 组件管理模块中可以存储多个组件, 如自定义组件(如 python 组件)、镜像组件、同容器组件、算法包组件、组件拉起镜像模板(简称为镜像模板)等。其中, 自定义组件为用户自主开发的组件。镜像组件为已经创建完成的镜像文件所对应的组件。算法包组件为包括多个代码块和依赖包的组件。组件拉起镜像模板用于为除了镜像组件以外的其他组件提供镜像文件, 以使得应用部署平台基于镜像文件加载组件, 并创建容器。并且, 组件管理模块还可以包括: 管理接口、部署接口、模型加载接口, 该管理接口用于对组件管理模块中的组件进行管理(如增加、删除等)。部署接口用于将组件管理模块中的组件进行部署。模型加载接口用于加载模型库中的模型。

10 运行管理模块用于对应用的运行进行管理。又例如, 运行管理模块可以用于部署画布中的组件, 即将画布中各组件以容器的形式部署, 使得容器运行组件, 以提供组件的服务。又例如, 运行管理模块可以用于部署组件管理模块中的组件。又例如, 运行管理模块可以对部署过程进行管理, 如部署的位置(如将应用部署在计算设备集群 201 中的一台服务器中)。又例如, 运行管理模块还可以实时监控应用的运行状态, 如应用的运行是否异常。又例如, 运行管理模块还可以记录应用的运行日志, 如在某一时刻运行发生异常。又例如, 运行管理模块还可以对画布中的组件进行配额管理, 如增加画布可编排组件数量。又例如, 运行管理模块还可以进行弹性扩容, 如增加 CPU 核数、升级网卡、升级硬盘等。

15 可选的, 该应用部署平台还可以包括模型管理模块。该模型管理模块可以用于存储多个模型。其中, 该模型可以为用户上传的模型, 或者该模型可以为通过应用部署平台训练得到的模型。

需要说明的是, 本申请提供的应用部署方法的执行主体可以为应用部署装置, 该应用部署装置可以为计算设备集群。该计算设备集群可以由多个服务器构成的一个或多个服务器集群。或者, 计算设备集群还可以是分布式集群。或者, 计算设备集群中的多个服务器可以是云端服务

25 器。本申请实施例对计算设备集群的具体实现方式不作限制。

同时, 该应用部署装置还可以为该计算设备集群的中央处理器(Central Processing Unit, CPU), 或者该计算设备集群中的用于开发应用的模块, 或者部署在该计算设备集群中的应用部署平台。或者, 该应用部署装置可以为计算设备集群中的一个服务器(如物理服务器或者云端服务器)。或者, 该应用部署装置还可以包括电子设备, 该电子设备登录有应用部署平台(或者应用部署软件)。以下实施例中以服务器和电子设备执行应用部署方法为例, 说明本申请实施例提供的应用部署方法。

30

在本申请实施例中, 应用部署方法可以分为步骤一和步骤二, 步骤一为用户通过电子设备显示的组件编排界面开发应用, 步骤二为服务器通过编排界面中配置完成的组件开发应用。

下面先对步骤一, 即用户通过电子设备显示的组件编排界面开发应用的过程进行介绍。在步骤一中, 电子设备可以登录部署在服务器的应用部署平台, 并显示组件编排界面。之后, 用户可以在电子设备输入操作, 以控制电子设备将多个组件放入一个组件中, 并发布应用。

35

本申请实施例提供一种应用部署方法, 如图 4 所示, 该应用部署方法可以包括:

S401、电子设备显示组件编排界面。

其中, 组件编排界面包括: 组件选择区和组件编排区, 组件选择区用于为用户提供待编排的组件, 组件选择区包括: 多个预设组件。组件编排区用于显示用户所选择的组件。组件编排区用于为用户提供编排组件的操作区。

40

在一种可能的设计中, 预设组件可以包括: 业务组件和/或逻辑组件。其中, 业务组件为可以执行业务功能的组件, 逻辑组件为可以执行逻辑功能的组件。

需要说明的是, 业务组件是指为了实现特定业务功能而设计和开发的 reusable 模块。业务组件可以独立存在, 并具有明确的输入、处理和输出, 从而能够完成特定的业务任务。逻辑组件通常封装了一组紧密相关的逻辑操作, 使得开发人员可以更方便地调用和管理这些操作, 从而提高代码的可维护性和可重用性。

45

在本申请实施例中，组件选择区还可以包括：同容器组件，该同容器组件用于指示与其自身关联的多个预设组件运行在同一容器中。其中，该同容器组件为应用部署平台预先开发的组件。

在一些实施例中，在电子设备显示组件编排界面之前，电子设备可以创建组件编排界面。电子设备可以接收用户的编排界面创建操作，响应于编排界面创建操作，电子设备可以显示组件编排界面。

示例性的，电子设备可以显示应用部署平台的首页界面，该首页界面包括应用开发选项，该编排界面创建操作可以为用户对应用开发选项的操作（如点击操作）。又例如，该编排界面创建操作可以为用户通过语音方式输入的操作，如用户输入“请创建组件编排界面”的语音指令。

可选的，电子设备可以接收界面配置操作，响应于界面配置操作，电子设备可以设置组件编排界面的配置信息。组件编排界面的配置信息可以包括：组件编排界面的名称、组件编排界面的描述信息等。

其中，组件编排界面的名称可以标识组件编排界面，组件编排界面的描述信息可以用于介绍该组件编排界面的功能、版本等信息。

示例性的，组件编排界面可以包括：名称输入框和描述信息输入框，响应于用户在名称输入框输入“应用 a 的编排界面”的操作，在描述信息输入框输入“用于实现计算功能”的操作，将该组件编排界面的名称设置为“应用 a 的编排界面”，描述信息为“用于实现计算功能”。

S402、响应于对同容器组件的选择操作，电子设备在组件编排区显示第一同容器组件。

其中，第一同容器组件与第一容器相对应。

应理解，第一同容器组件与第一容器相对应是指，第一同容器组件与第一容器所对应的第一镜像文件相对应，且通过第一镜像文件创建第一容器后，该第一容器能够运行第一同容器组件。该第一同容器组件为逻辑组件。

在一种可能的实现方式中，响应于对同容器组件的选择操作，电子设备在组件编排区以虚线框或实线框的形式显示第一同容器组件。

需要说明的是，本申请实施例对选择操作不作限定。例如，该选择操作可以为点击操作。又例如，该选择操作可以为拖、拉、拽等操作。本申请实施例对虚线框或实线框的形状不作限定。例如，虚线框可以为长方形、正方形、圆形等形状显示在组件编排区中。

在一示例性的，如图 5 所示，画布 501（即组件编排界面）可以包括：组件选择区 502（如组件树）和组件编排区 503。组件选择区 502 中可以包括：平台组件（如输入组件、输出组件、同容器组件（即同容器组件，如 DEPLOY-TOGETHER）等）、业务组件、应用组件、自定义组件等。响应于用户对同容器组件 504 的拖拽操作，电子设备在画布 501 中显示同容器组件 504。

可选的，响应于多次对同容器组件的选择操作，电子设备可以在组件编排区显示多个第一同容器组件。

S403、响应于对多个第一预设组件的选择操作，电子设备在第一同容器组件中显示多个第一预设组件。

其中，第一预设组件为多个预设组件中任一组件，多个第一预设组件与第一容器相对应。

应理解，多个第一预设组件与第一容器相对应是指，多个第一预设组件与第一容器所对应的第一镜像文件相对应，且通过第一镜像文件创建第一容器后，该第一容器能够运行多个第一预设组件。也就是说，第一容器可以运行多个第一预设组件。

这样一来，通过多个第一预设组件共用一个容器，无需为每个第一预设组件创建一个容器，能够减少容器占用的资源量，以提升资源的利用率。

需要说明的是，本申请实施例对第一预设组件的数量不作限定。例如，第一预设组件的数量可以为 3 个、5 个、8 个等。本申请实施例对第一预设组件的类型不作限定。例如，该第一预设组件可以为平台组件、业务组件、自定义组件等。

示例性的，结合图 5，如图 6 所示，响应于用户对业务组件 A、业务组件 B、自定义组件 A（上述组件均在组件选择区中，图 6 中未示出）的拖拽操作，电子设备在同容器组件 504 中显示业务组件 A、业务组件 B、自定义组件 A。

在一些实施例中，在组件编排区显示多个第一同容器组件之后，电子设备可以接收对多个第

一预设组件的编排操作。响应于对多个第一预设组件的编排操作，电子设备可以建立多个第一预设组件之间的连接关系。并且，电子设备可以显示多个第一预设组件之间的连接关系。

示例性的，结合图 6 可知，业务组件 A 与业务组件 B 之间具有连接关系，业务组件 B 与自定义组件 A 之间具有连接关系，业务组件 A 与业务组件 B 之间存在连线（连线可以有箭头，也可以没有箭头），业务组件 B 与自定义组件 A 之间存在连线。

需要说明的是，该编排操作可以包括：组件间的连接操作、组件间的连接关系的调整操作（即改变组件之间的连接关系）。本申请实施例对编排操作的实现方式不作限定。例如，用户可以通过对组件本身操作实现对组件的编排操作，如点击组件编排区域中组件的任一位置。又例如，用户可以通过对组件的连接点进行操作实现对组件的编排操作，如点击组件的连接点。又例如，用户可以通过输入连接关系实现对组件的编排操作，如组件编排区存在输入框，该输入框用于设置组件之间的连接关系（如在输入框输入“建立业务组件 A 与业务组件 B 之间的关系”）。

在一种可能的实现方式中，多个第一预设组件可以包括：第一子组件、第二子组件、第三子组件。响应于用户连接第一子组件和第二子组件的连接操作，电子设备可以连接第一子组件和第二子组件，并显示第一子组件和第二子组件之间的连线。响应于用户连接第二子组件和第三子组件的连接操作，电子设备可以连接第二子组件和第三子组件，并显示第二子组件和第三子组件之间的连线。

本申请中，存在连线的组件具备连接关系，该连接关系用于指示组件之间的参数传递。示例性的，结合图 6，业务组件 A 与业务组件 B 之间存在连线，业务组件 B 与自定义组件 A 之间存在连线。业务组件 A 的输出参数为业务组件 B 的输入参数，业务组件 B 的输出参数为自定义组件 A 的输入参数。

可选的，响应于用户对多个第一预设组件之间的连接关系的调整操作，电子设备可以更新多个第一预设组件的连接关系。

应理解，组件间的连接关系用于指示组件之间的参数传递。

可以理解的是，通过对多个第一预设组件的编排操作，电子设备可以建立多个第一预设组件之间的连接关系。如此，可以保障多个组件在一个容器中运行时，组件之间的运行顺序和容器内数据的传递方向，使得容器能够正常运行。

S404、响应于应用发布操作，电子设备向服务器发送应用发布请求。

其中，应用发布请求用于指示根据组件编排区中的全部组件生成目标应用。应用发布请求可以包括：组件编排区中全部组件的组件标识、全部组件中各个组件之间的连接关系。

在一种可能的设计中，组件编排界面可以包括发布选项，该发布选项用于触发将组件编排区中的全部组件生成应用。应用发布操作为用户对发布选项的操作。

在本申请实施例中，组件编排区中的全部组件可以包括：第一同容器组件和第一同容器组件中的多个第一预设组件。

需要说明的是，第一同容器组件中的多个第一预设组件是指，在组件编排区中位于第一同容器组件所在区域中的组件（如图 6 中，位于同容器组件 504 中的业务组件 A、业务组件 B、自定义组件 A）。

可选的，组件编排区中的全部组件还可以包括：至少一个第二预设组件，第二预设组件为多个预设组件中任一组件，第二预设组件在第一同容器组件外。也就是说，第二预设组件为组件编排区中不在第一同容器组件内的预设组件。

在一些实施例中，在电子设备接收用户的应用发布操作之前，电子设备可以接收用户对第二预设组件的选择操作。响应于对至少一个第二预设组件的选择操作，在组件编排区中显示至少一个第二预设组件。之后，响应于对第二预设组件和多个第一预设组件中边缘组件的编排操作，建立第二预设组件与边缘组件之间的连接关系，边缘组件为多个第一预设组件中与第二预设组件存在连接关系的组件。

也就是说，第一同容器组件内的第一预设组件可以通过边缘组件与第一同容器组件以外的组件建立连接关系。

需要说明的是，本申请实施例对边缘组件的数量不作限定。例如，边缘组件可以为 1 个、3

个、5个等。

在一种可能的设计中，边缘组件可以包括：边缘输入组件和/或边缘输出组件。边缘输入组件用于接收第二预设组件传递的参数，边缘输出组件用于向第二预设组件传递参数。

也就是说，边缘输入组件的输入源为与该边缘输入组件连接的第二预设组件，该第二预设组件的输出参数为边缘输入组件的输入参数。边缘输出组件的输出源为与该边缘输出组件连接的第二预设组件，该第二预设组件的输入参数为边缘输出组件的输出参数。

示例性的，结合图6，如图7所示，响应于用户对输入组件和输出组件的拖拽操作，电子设备在画布501中显示输入组件和输出组件。并且，响应于用户对输入组件与业务组件A之间的连接操作，电子设备在画布501中显示输入组件与业务组件A（即边缘输入组件）之间的连线。响应于用户对输出组件与自定义组件A之间的连接操作，电子设备在画布501中显示输出组件与自定义组件A（即边缘输出组件）之间的连线。

可以理解的是，通过对第二预设组件与多个第一预设组件中边缘组件的编排操作，电子设备可以建立第二预设组件与边缘组件之间的连接关系。如此，可以保障第二预设组件所对应的容器与第一容器之间能够正常访问，进而保障应用能够正常运行。

可选的，响应于对不同第二预设组件的编排操作，电子设备可以建立不同第二预设组件之间的连接关系。

基于上述技术方案，电子设备在显示组件编排界面后，可以响应于对同容器组件和多个第一预设组件的选择操作，在组件编排区显示第一同容器组件，并在第一同容器组件中显示多个第一预设组件。也就是说，多个组件均位于一个组件中。然后，电子设备向服务器发送应用发布请求时，服务器可以根据组件编排区中的全部组件生成目标应用。由于第一同容器组件和第一预设组件均对应第一容器，服务器在部署目标应用时，可以将多个第一预设组件运行在一个第一容器中。如此，可以减少部署容器的数量，从而减少容器对资源的占用。

在一些实施例中，在电子设备接收应用发布操作之前，电子设备可以接收用户的编排保存操作。响应于编排保存操作，电子设备可以向服务器发送编排保存消息，该编排保存消息用于指示保存组件编排界面中的全部组件、组件之间的连接关系以及组件的配置信息。之后，服务器可以保存组件编排界面中的全部组件、组件之间的连接关系以及组件的配置信息。

可以理解的是，通过编排保存操作，服务器可以及时保存用户的编排记录，便于用户后续继续编排，以提高应用开发的效率。

以上是对组件编排界面中编排组件的过程进行介绍。下面对组件编排界面中配置组件的过程进行介绍。

在一些实施例中，在电子设备接收用户的应用发布操作之前，响应于对组件的配置操作，电子设备可以配置组件的配置信息。

在本申请实施例中，应用发布请求还可以包括：组件编排界面中全部组件的配置信息。需要说明的是，本申请实施例对组件的配置信息不作限定。例如，组件的配置信息可以包括：组件标识（如组件id、组件名称）、组件版本、组件服务、输入参数、输出参数、部署参数等。其中，组件id和组件名称是唯一的，组件服务用于指示组件提供的服务，输入参数为组件接收的参数，输出参数为组件向其他组件传递的参数，部署参数用于指示部署组件对应的容器所需要的资源量。其中，部署参数可以包括：规格名称、虚拟机规格（例如1U2G、2U4G等，其中“1U2U”可以是指一个CPU和2GB内存。或者，1单元（unit，U）是一种表示服务器外部尺寸的高度或厚度的单位，如1U=4.445厘米）、磁盘大小（例如32GB、100GB等）等。

需要说明的是，上述配置信息中的部分信息（如组件id、组件名称、组件版本等）可以为用户填写的数据，也可以为应用部署平台预先设置的数据，本申请实施例对此不作限定。

在一种可能的实现方式中，响应于对第一同容器组件的配置操作，电子设备可以在组件配置区设置目标资源量，目标资源量为第一同容器组件对应的第一容器所需的资源量。其中，第一同容器组件的配置信息包括部署参数。

应理解，由于该第一容器运行的组件包括多个第一预设组件，该目标资源量实际为运行多个第一预设组件所需要的资源量。

示例性的，结合图 7，响应于用户对同容器组件 504 的操作，电子设备在组件编排界面显示组件配置区 701，组件配置区 701 包括部署参数。之后，响应于用户在组件配置区 701 输入部署参数的操作，电子设备可以将同容器组件 504 的部署参数设置为 CPU 为 4cores，内存为 4096 兆字节(MB)。

可以理解的是，由于第一容器对应多个第一预设组件，电子设备在组件配置区设置目标资源量，可以保障第一容器能够分配足够多的资源量，以保障第一容器运行多个第一预设组件。并且，由于多个第一预设组件均运行在第一同容器组件对应的第一容器中，用户无需对每个第一预设组件进行部署参数的配置，可以减少用户的配置过程。

可选的，响应于对组件编排区中第一预设组件的配置操作，电子设备可以配置每个第一预设组件的配置信息。

示例性的，结合图 7，如图 8 所示，响应于对业务组件 A 的配置操作，电子设备可以显示组件配置区 801。其中，组件配置区 801 包括：输入配置、输出配置。输入配置包括：输入参数的配置、输入文件的配置（包括配置文件和模型文件）、输入的历史数据、服务参数等。例如，输入参数的配置中，组件 A 的输入参数包括：A、B、C，A 的映射参数为 a（即输入组件的输出参数 a），B 的映射参数为 b（即输入组件的输出参数 b），C 的映射参数为 c（即输入组件的输出参数 c）。输出配置包括：输出参数的配置（如 A1、B1、C1）、输出的历史数据。可选的，组件配置区 801 还包括部署参数的配置。

需要说明的是，由于业务组件 A 在同容器组件 504 中，业务组件 A 最终会在同容器组件 504 对应的容器中运行。因此，组件配置区 801 中的部署配置是无效信息。

可选的，第一预设组件的配置信息中不包括部署参数。

可选的，响应于对组件编排区中第二预设组件的配置操作，电子设备可以配置每个第二预设组件的配置信息。

以上是对电子设备实现步骤一（即用户通过电子设备显示的组件编排界面开发应用）过程的介绍。下面对步骤二（即服务器通过编排界面中配置完成的组件开发应用）的过程进行介绍。

在一些实施例中，在接收应用部署请求之前，响应于接收到的应用发布请求，服务器根据组件编排区中的全部组件，生成目标应用，全部组件包括同容器组件和多个第一预设组件。

下面对服务器根据组件编排区中的全部组件，生成目标应用的过程进行介绍。在服务器根据组件编排区中的全部组件，生成目标应用的过程中，服务器需要分别执行以下三个过程得到目标应用。三个过程包括：过程一（确定全部组件之间的关系）、过程二（融合第一同容器组件中的多个第一预设组件）、过程三（对边缘组件传递的参数进行处理）。

需要说明的是，本申请实施例对服务器执行过程一、过程二、过程三的顺序不作限定。例如，服务器可以先执行过程一，再执行过程二，然后执行过程三。又例如，服务器可以先执行过程二，再执行过程一，然后执行过程三。又例如，服务器可以同时执行过程一、过程二、过程三。

下面先对服务器执行过程一，即服务器确定全部组件之间的关系的过程进行介绍。

在本申请实施例中，服务器可以从组件编排界面中获取组件编排区中全部组件的连接关系。之后，服务器可以根据组件编排界面中全部组件的连接关系，确定各个组件之间的关系。

需要说明的是，各个组件之间的关系是指，每两个组件之间传递参数时的身份（输入源或者输出源）。例如，结合图 6 可知，在业务组件 A 与业务组件 B 的关系中，业务组件 A 为输入源，业务组件 B 为输出源。在业务组件 B 与自定义组件 A 的关系中，业务组件 B 为输入源，自定义组件 A 为输出源。

在一种可能的设计中，各个组件之间的关系包括：多个第一预设组件之间的关系。

可选的，各个组件之间的关系还可以包括：第二预设组件与边缘组件之间的关系、第二预设组件与第二预设组件之间的关系。

可以理解的是，服务器确定全部组件的关系，能够保障应用运行时组件间的相互协调，使得应用能够正常运行。

在对过程一进行介绍之后，下面对过程二，即服务器融合第一同容器组件中的多个第一预设

组件的过程进行介绍。

本申请实施例提供一种应用部署方法，如图9所示，该应用部署方法可以包括：服务器可以解压每个第一预设组件，得到每个第一预设组件对应的解压文件。之后，服务器将多个第一预设组件对应的全部解压文件进行压缩，得到融合组件。然后，服务器可以根据融合组件，生成目标应用。

需要说明的是，本申请实施例对解压文件不作限定。例如，第一预设组件对应的解压文件可以包括：静态资源（如HTML、CSS、JavaScript文件）、应用程序代码（如Python、Node.js代码）等。并且，在本申请实施例中，由于多个第一预设组件融合为了融合组件，则第二预设组件与边缘组件之间的关系具体为：第二预设组件与融合组件之间的关系。

需要说明的是，具体对于服务器根据融合组件生成目标应用的介绍，可以参考常规技术中，组件编排平台响应发布应用的操作生成应用的过程，此处不予赘述。例如，发布后的目标应用可以包括以下至少一项：融合组件的软件包、应用程序接口（application programming interface, API）、镜像文件、镜像配置文件等。其中，API可以完成容器间的相同通信（如程序间方法的调用）。镜像配置文件是指镜像文件启动时使用的配置信息，该配置信息可以包括：数据库链接的地址、用户名、密码，中间件的连接信息，日志记录服务器的连接信息、日志记录格式等。

可以理解的是，服务器通过将多个第一预设组件进行融合，得到融合组件，使得服务器部署第一容器时，只需要加载融合组件，便可以使得第一容器运行多个第一预设组件，以减少部署的容器数量。

在本申请实施例中，在生成融合组件之后，服务器可以将调用的第一预设组件的接口更新为融合组件的接口。

示例性的，第一预设组件的接口为接口a，融合组件的接口为接口b。在生成融合组件之前，组件1调用第一预设组件时需要调用接口a。在生成融合组件之后，组件1调用第一预设组件时需要调用接口b。

这样一来，其他组件调用第一预设组件时，可以通过调用融合组件以实现第一预设组件的功能，并完成参数的传递。

在对过程二进行介绍之后，下面对过程三，即服务器对边缘组件传递的参数进行处理的过程进行介绍。

在本申请实施例中，服务器可以根据组件编排区中的全部组件和全部组件中各个组件之间的连接关系，生成目标应用。

在一种可能的设计中，全部组件包括：多个第一预设组件，全部组件中各个组件之间的连接关系包括：第一预设组件之间的连接关系。服务器可以根据多个第一预设组件之间的连接关系，确定多个第一预设组件之间的关系。然后，服务器可以根据多个第一预设组件、多个第一预设组件和第一预设组件之间的关系，生成目标应用。

可选的，全部组件还可以包括：至少一个第二预设组件。全部组件中各个组件之间的连接关系还可以包括：第二预设组件与边缘组件之间的连接关系。

需要说明的是，由于边缘组件包括：边缘输入组件和/或边缘输出组件，则说明第二预设组件中可以包括：边缘组件的输入源和/或边缘组件的输出源。

在一种可能的实现方式中，至少一个第二预设组件包括：至少一个第三预设组件，边缘组件包括：边缘输入组件，第三预设组件为向边缘输入组件传递参数的组件。服务器可以获取每个第三预设组件的输出参数。之后，服务器可以将每个第三预设组件的输出参数进行聚合，得到第一同容器组件的输入参数集合。然后，服务器可以根据每个第三预设组件与边缘输入组件之间的连接关系，建立输入参数集合中各个参数与边缘输入组件的输入参数之间的映射关系。

也就是说，该输入参数集合为输入第一同容器组件的全部参数。应理解，本申请中多个第一预设组件均在第一同容器组件中，且融合组件是通过多个第一预设组件生成的。因此，输入第一同容器组件的参数即为输入融合组件的参数。

示例性的，假如边缘输入组件包括：组件a和组件b，组件a的输入参数为a1和a2，组件b的输入参数为b1，至少一个第三预设组件包括：组件c和组件d，组件c的输出参数为c1和

c2, 组件 d 的输出参数为 d1。其中, 组件 c 与组件 a 具有连接关系, 组件 d 与组件 b 具有连接关系。则如表 1 所示, 其示出了输入参数集合与边缘输入组件的输入参数之间的映射关系。

表 1

输入参数集合	边缘输入组件的输入参数
c1	a1
c2	a2
d1	b1

也就是说, 组件 c 的输出参数 c1 与组件 a 的输入参数 a1 具有映射关系, 组件 c 的输出参数 c2 与组件 a 的输入参数 a2 具有映射关系, 组件 d 的输出参数 d1 与组件 b 的输入参数 b1 具有映射关系。

可以理解的是, 通过建立输入参数集合中各个参数与边缘输入组件的输入参数之间的映射关系, 可以保障应用运行时, 容器间能够正常传递参数。

在另一种可能的实现方式中, 至少一个第二预设组件包括: 至少一个第四预设组件, 边缘组件还包括: 边缘输出组件, 边缘输出组件为向第四预设组件传递参数的组件。服务器可以获取每个边缘输出组件的输出参数。之后, 服务器可以将每个边缘输出组件的输出参数进行聚合, 得到第一同容器组件的输出参数集合。然后, 根据每个第四预设组件与边缘输出组件之间的连接关系, 建立输出参数集合中各个参数与第四预设组件的输入参数之间的映射关系。

也就是说, 该输出参数集合为第一同容器组件输出的全部参数。应理解, 本申请中多个第一预设组件均在第一同容器组件中, 且融合组件是通过多个第一预设组件生成的。因此, 第一同容器组件输出的参数即为融合组件输出的参数。

示例性的, 假如边缘输出组件包括: 组件 e 和组件 f, 组件 e 的输出参数为 e1 和 e2, 组件 f 的输出参数为 f1, 至少一个第四预设组件包括: 组件 g 和组件 h, 组件 g 的输入参数为 g1 和 g2, 组件 h 的输入参数为 h1。其中, 组件 e 与组件 g 具有连接关系, 组件 f 与组件 h 具有连接关系。则如表 2 所示, 其示出了输出参数集合与第四预设组件的输入参数之间的映射关系。

表 2

输出参数集合	第四预设组件的输入参数
e1	g1
e2	g2
f1	h1

也就是说, 组件 e 的输出参数 e1 与组件 g 的输入参数 g1 具有映射关系, 组件 e 的输出参数 e2 与组件 g 的输入参数 g2 具有映射关系, 组件 f 的输出参数 f1 与组件 h 的输入参数 h1 具有映射关系。

可以理解的是, 通过建立输出参数集合中各个参数与第四预设组件的输入参数之间的映射关系, 可以保障应用运行时, 容器间能够正常传递参数。

在另一种可能的实现方式中, 多个第二预设组件包括: 至少一个第三预设组件和至少一个第四预设组件, 边缘组件包括: 边缘输入组件和边缘输出组件。

下面结合具体示例对过程三进行介绍。

如图 10 所示, 目标应用由组件一、组件二、组件三、组件四构成, 组件二和组件三对应同一个容器。服务器可以将组件一的输出参数 1 和输出参数 2 进行输入聚合, 得到包括输出参数 1 和输出参数 2 的输入集合。之后, 服务器可以建立输出参数 1 与组件二的输入参数 1 之间的映射关系、输出参数 2 与组件三的输入参数 2 之间的映射关系 (即输入展开)。并且, 服务器可以将组件二的输出参数 3 和组件三的输出参数 4 进行输出聚合, 得到包括输出参数 3 和输出参数 4 的输出集合。之后, 服务器可以建立输出参数 3 与组件四的输入参数 3 之间的映射关系、输出参数 4 与组件四的输入参数 4 之间的映射关系 (即输出展开)。

以上是对应用开发过程中的步骤一 (用户通过电子设备显示的组件编排界面开发应用) 和步骤二 (服务器通过编排界面中配置完成的组件开发应用) 的介绍。在应用开发完成后, 用户可以通过电子设备, 控制服务器部署已经发布的应用。下面对部署应用的过程进行介绍。

本申请实施例提供一种应用部署方法，如图 11 所示，该应用部署方法可以包括：

S1101、服务器提供组件编排界面。

其中，组件编排界面包括：组件选择区和组件编排区。组件选择区包括：同容器组件和多个预设组件，同容器组件用于指示与其自身关联的多个预设组件运行在同一容器中，组件编排区用于为用户提供编排组件的操作区，多个预设组件包括业务组件和/或逻辑组件。

S1102、响应于用户在组件编排区的操作，服务器在组件编排区中创建第一同容器组件，并确定与第一同容器组件关联的多个第一预设组件。

其中，多个预设组件包含多个第一预设组件。

在一种可能的实现方式中，响应于用户在组件编排区的操作，电子设备可以向服务器发送组件编排请求。其中，该用户在组件编排区的操作可以包括 S402 中对同容器组件的选择操作和 S403 中对多个第一预设组件的选择操作。

可选的，该在组件编排区的操作还可以包括对多个组件（如多个第一预设组件、和/或、多个第二预设组件）的编排操作。

S1103、响应于接收到的应用部署请求，服务器部署包含多个第一预设组件的目标应用。

其中，多个第一预设组件运行在第一同容器组件对应的第一容器中。

在一些实施例中，电子设备可以接收用户的应用部署操作。响应于应用部署操作，电子设备可以向服务器发送应用部署请求，应用部署请求用于指示部署目标应用。其中，部署后的目标应用包括：第一容器，第一容器用于运行多个第一预设组件。

应理解，第一容器运行多个第一预设组件是指，第一容器运行融合组件。

在一示例性的，组件编排界面还包括部署选项，电子设备可以接收用户对部署选项的选择操作，向服务器发送应用部署请求。

在另一示例性的，电子设备可以显示应用列表，该应用列表包括已经发布的的应用的应用标识。之后，响应于用户对目标应用的部署操作，电子设备向服务器发送应用部署请求。

如图 12 所示，电子设备可以显示应用部署界面 1201，该应用部署界面 1201 包括：应用标识（如名称）、版本、状态、创建时间、描述信息、操作选项（如部署、详情、删除）。响应于对应用 a 的部署选项的操作，电子设备可以指示部署应用 a。

在本申请实施例中，服务器可以获取应用部署请求，应用部署请求用于指示部署目标应用。之后，服务器可以根据第一同容器组件和多个第一预设组件，部署第一容器，第一容器用于运行多个第一预设组件。其中，第一容器还用于运行第一同容器组件。

在一种可能的实现方式中，服务器可以根据第一同容器组件、融合组件和第一同容器组件对应的目标镜像文件，创建第一容器。之后，服务器可以将第一容器部署在目标环境中。

需要说明的是，本申请实施例对目标环境不作限定。例如，目标环境可以为该服务器。又例如，该目标环境可以为服务器所在的计算设备集群中任一服务器中。

在一种可能的设计中，服务器可以从组件编排界面中获取目标资源量的大小，目标资源量为第一同容器组件对应的第一容器所需的资源量。之后，服务器可以为第一容器分配目标资源量。

示例性的，结合图 7，该目标资源量可以为 CPU 为 4cores，内存为 4096 兆字节(MB)。

可以理解的是，由于第一同容器组件中包括多个第一预设组件，服务器在创建第一容器时，可以基于第一同容器组件中的部署参数，为第一容器分配目标资源量，以保障第一容器能够运行多个第一预设组件。并且，由于多个第一预设组件运行在一个第一容器中，无需服务器为每个组件的容器单独分配启动时的资源量，从而减少了容器占用的资源数量。

在一些实施例中，部署后的目标应用还包括：每个第二预设组件对应的第二容器，第二容器用于运行对应的第二预设组件。

需要说明的是，本申请实施例对第一容器和第二容器的部署环境不作限定。例如，第一容器和第二容器可以部署在同一个部署环境中（如同一台服务器）。又例如，第一容器和第二容器可以部署在不同部署环境中。

基于上述技术方案，可以通过用户在组件编排区的操作，在组件编排区中创建第一同容器组件，并确定与第一同容器组件关联的多个第一预设组件。如此，在部署应用时，第一同容器组件

和多个第一预设组件均可以运行在同一容器中。这样一来，可以通过一个容器加载多个组件，减少了容器的部署数量，从而减少容器占用的资源量。并且，由于一个容器可以运行多个组件，使得多个组件可以在容器中通过内存进行访问，从而提高组件间的访问效率，进而提升应用的流畅度。

5 以上为对部署目标应用过程的介绍。下面对运行目标应用的过程进行介绍。

在一些实施例中，在运行第一容器的情况下，多个第一预设组件之间可以通过目标数据结构传递参数，目标数据结构为在内存中使用的数据结构。

应理解，在容器中传递参数是在同一内存中传递参数。因此，传递的参数所使用的数据结构需要是在内存中使用的数据结构。本申请实施例对目标数据结构不作限定。例如，目标数据结构可以包括但不限于：队列、数组、链表、栈等。以下以目标数据结构为队列为例，对运行第一容器的过程进行介绍。

如图 13 所示，第一容器中运行有初始化队列组件、视频抽帧组件、推理组件和后处理组件。初始化队列组件可以初始化 2 个队列（原始图片队列和推理结果队列），并将 2 个队列作为输出参数，传递至视频抽帧组件和推理组件使用。视频抽帧组件可以获取视频流，并从视频流中抽帧得到多个图片，存入原始图片队列中。推理组件可以读取原始图片队列中存入的图片，对图片进行推理得到推理结果，并将推理结果存入推理结果队列。后处理组件可以从推理结果组件中读取推理结果，并对推理结果进行后处理，得到后处理结果。然后，后处理组件可以将后处理结果反馈至用户。

20 这样一来，由于内存中传递参数的速度较快，多个第一预设组件之间通过目标数据结构在内存中传递参数，能够提高数据的传递效率，降低时延，进而提高应用的运行速度。

上述主要从电子设备的角度对本申请实施例提供的方案进行了介绍。可以理解的是，电子设备为了实现上述功能，其包含了执行各个功能相应的硬件结构和/或软件模块。本领域技术人员应该很容易意识到，结合本申请所公开的实施例描述的各示例的一种应用部署方法步骤，本申请能够以硬件或硬件和计算机软件的结合形式来实现。某个功能究竟以硬件还是电子设备软件驱动硬件的方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

本申请实施例可以根据上述方法示例对应用部署装置进行功能模块或者功能单元的划分，例如，可以对应各个功能划分各个功能模块或者功能单元，也可以将两个或两个以上的功能集成在一个处理模块中。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或者功能单元的形式实现。其中，本申请实施例中对模块或者单元的划分是示意性的，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式。

本申请还提供一种应用部署装置，如图 14 所示，包括：界面提供模块、处理模块、获取模块。

35 界面提供模块，用于提供组件编排界面，组件编排界面包括：组件选择区和组件编排区，组件选择区包括：同容器组件和多个预设组件，同容器组件用于指示与其自身关联的多个预设组件运行在同一容器中，组件编排区用于为用户提供编排组件的操作区，多个预设组件包括业务组件和/或逻辑组件。

40 处理模块，用于响应于用户在组件编排区的操作，在组件编排区中创建第一同容器组件，并确定与第一同容器组件关联的多个第一预设组件，多个预设组件包含多个第一预设组件。处理模块，用于响应于接收到的应用部署请求，部署包含多个第一预设组件的目标应用，其中，多个第一预设组件运行在第一同容器组件对应的第一容器中。

在一种可能的设计方式中，处理模块，还用于响应于接收到的应用发布请求，根据组件编排区中的全部组件，生成目标应用，应用发布请求用于指示根据组件编排区中的全部组件生成目标应用，全部组件包括同容器组件和多个第一预设组件。

45 在另一种可能的设计方式中，组件编排界面还包括：组件配置区。获取模块，用于从第一同容器组件对应的组件配置区中获取目标资源量的大小，目标资源量为第一同容器组件对应的第一容器所需的资源量。处理模块，用于为第一容器分配目标资源量。

在另一种可能的设计方式中，处理模块，用于解压每个第一预设组件，得到每个第一预设组件对应的解压文件。处理模块，用于将多个第一预设组件对应的全部解压文件进行压缩，得到融合组件，并根据融合组件，生成目标应用。

5 在另一种可能的设计方式中，处理模块，用于将调用的第一预设组件的接口均更新为调用融合组件的接口。

在另一种可能的设计方式中，处理模块，用于根据组件编排区中的全部组件和全部组件中各个组件之间的连接关系，生成目标应用。

在另一种可能的设计方式中，全部组件包括：多个第一预设组件；全部组件中各个组件之间的连接关系包括：第一预设组件之间的连接关系。

10 在另一种可能的设计方式中，全部组件还包括：至少一个第二预设组件，第二预设组件为多个预设组件中任一组件，第二预设组件在第一同容器组件外；全部组件中各个组件之间的连接关系还包括：第二预设组件与边缘组件之间的连接关系，边缘组件为多个第一预设组件中与第二预设组件存在连接关系的组件。

15 在另一种可能的设计方式中，至少一个第二预设组件包括：至少一个第三预设组件，边缘组件包括：边缘输入组件，第三预设组件为向边缘输入组件传递参数的组件。处理模块，用于获取每个第三预设组件的输出参数。处理模块，用于将每个第三预设组件的输出参数进行聚合，得到第一同容器组件的输入参数集合。处理模块，用于根据每个第三预设组件与边缘输入组件之间的连接关系，建立输入参数集合中各个参数与边缘输入组件的输入参数之间的映射关系。

20 在另一种可能的设计方式中，至少一个第二预设组件还包括：至少一个第四预设组件，边缘组件还包括：边缘输出组件，边缘输出组件为向第四预设组件传递参数的组件。处理模块，用于获取每个边缘输出组件的输出参数。处理模块，用于将每个边缘输出组件的输出参数进行聚合，得到第一同容器组件的输出参数集合。处理模块，用于根据每个第四预设组件与边缘输出组件之间的连接关系，建立输出参数集合中各个参数与第四预设组件的输入参数之间的映射关系。

25 在另一种可能的设计方式中，在运行第一容器的情况下，多个第一预设组件之间通过目标数据结构传递参数，目标数据结构为在内存中使用的数据结构。

其中，处理模块、获取模块和界面提供模块均可以通过软件实现，或者可以通过硬件实现。示例性的，接下来以处理模块为例，介绍处理模块的实现方式。类似的，获取模块和界面提供模块的实现方式可以参考处理模块的实现方式。

30 模块作为软件功能单元的一种举例，处理模块可以包括运行在计算实例上的代码。其中，计算实例可以包括物理主机（计算设备）、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，处理模块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。

35 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。

40 模块作为硬件功能单元的一种举例，处理模块可以包括至少一个计算设备，如服务器等。或者，处理模块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。

45 处理模块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。处理模块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，处

理模块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，所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。

需要说明的是，在其他实施例中，处理模块可以用于执行应用部署方法中的任意步骤，获取模块可以用于执行应用部署方法中的任意步骤，界面提供模块可以用于执行应用部署方法中的任意步骤，处理模块、获取模块、以及界面提供模块负责实现的步骤可根据需要指定，通过处理模块、获取模块、以及界面提供模块分别实现应用部署方法中不同的步骤来实现应用部署装置的全部功能。

模块作为软件功能单元的一种举例，应用部署装置可以包括运行在计算实例上的代码。其中，计算实例可以是物理主机（计算设备）、虚拟机、容器等计算设备中的至少一种。进一步地，上述计算设备可以是一台或者多台。例如，应用部署装置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，用于运行该应用程序的多个主机/虚拟机/容器可以分布在相同的 region 中，也可以分布在不同的 region 中。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 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 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。

模块作为硬件功能单元的一种举例，应用部署装置可以包括至少一个计算设备，如服务器等。或者，应用部署装置也可以是利用 ASIC 实现、或 PLD 实现的设备等。其中，上述 PLD 可以是 CPLD、FPGA、GAL 或其任意组合实现。

应用部署装置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。应用部署装置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，应用部署装置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，所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。

本申请还提供一种计算设备 150。如图 15 所示，计算设备 150 包括：总线 1502、处理器 1504、存储器 1506 和通信接口 1508。处理器 1504、存储器 1506 和通信接口 1508 之间通过总线 1502 通信。计算设备 150 可以是服务器或终端设备。应理解，本申请不限定计算设备 150 中的处理器、存储器的个数。

总线 1502 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 15 中仅用一条线表示，但并不表示仅有一根总线或一种类型的总线。总线 1504 可包括在计算设备 150 各个部件（例如，存储器 1506、处理器 1504、通信接口 1508）之间传送信息的通路。

处理器 1504 可以包括中央处理器（central processing unit，CPU）、图形处理器（graphics processing unit，GPU）、微处理器（micro processor，MP）或者数字信号处理器（digital signal processor，DSP）等处理器中的任意一种或多种。

存储器 1506 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）。处理器 1504 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、机械硬盘（hard disk drive，HDD）或固态硬盘（solid state drive，SSD）。

存储器 1506 中存储有可执行的程序代码，处理器 1504 执行该可执行的程序代码以分别实现前述处理模块、获取模块和界面提供模块的功能，从而实现应用部署方法。也即，存储器 1506 上存有用于执行应用部署方法的指令。

通信接口 1503 使用例如但不限于网络接口卡、收发器一类的收发模块，来实现计算设备 150 与其他设备或通信网络之间的通信。

本申请实施例还提供了一种计算设备集群。该计算设备集群包括至少一台计算设备。该计算设备可以是服务器，例如是中心服务器、边缘服务器，或者是本地数据中心中的本地服务器。在一些实施例中，计算设备也可以是台式机、笔记本电脑或者智能手机等终端设备。

如图 16 所示，所述计算设备集群包括至少一个计算设备 150。计算设备集群中的一个或多个计算设备 150 中的存储器 1506 中可以存有相同的用于执行应用部署方法的指令。

在一些可能的实现方式中，该计算设备集群中的一个或多个计算设备 150 的存储器 1506 中也可以分别存有助于执行应用部署方法的部分指令。换言之，一个或多个计算设备 150 的组合可以共同执行用于执行应用部署方法的指令。

需要说明的是，计算设备集群中的不同的计算设备 150 中的存储器 1506 可以存储不同的指令，分别用于执行应用部署装置的部分功能。也即，不同的计算设备 150 中的存储器 1506 存储的指令可以实现处理模块、获取模块和界面提供模块中的一个或多个模块的功能。

在一些可能的实现方式中，计算设备集群中的一个或多个计算设备可以通过网络连接。其中，所述网络可以是广域网或局域网等等。图 17 示出了一种可能的实现方式。如图 17 所示，两个计算设备 150A 和 150B 之间通过网络进行连接。具体地，通过各个计算设备中的通信接口与所述网络进行连接。在这一类可能的实现方式中，计算设备 150A 中的存储器 1506 中存有执行处理模块的功能的指令。同时，计算设备 150B 中的存储器 1506 中存有执行获取模块和界面提供模块的功能的指令。

图 17 所示的计算设备集群之间的连接方式可以是考虑到本申请提供的应用部署方法需要大量地存储数据和大量的计算资源，因此考虑将获取模块和界面提供模块实现的功能交由计算设备 150B 执行。

应理解，图 17 中示出的计算设备 150A 的功能也可以由多个计算设备 150 完成。同样，计算设备 150B 的功能也可以由多个计算设备 150 完成。

本申请实施例还提供了另一种计算设备集群。该计算设备集群中各计算设备之间的连接关系可以类似的参考图 16 和图 17 所述计算设备集群的连接方式。不同的是，该计算设备集群中的一个或多个计算设备 150 中的存储器 1506 中可以存有相同的用于执行应用部署方法的指令。

在一些可能的实现方式中，该计算设备集群中的一个或多个计算设备 150 的存储器 1506 中也可以分别存有助于执行应用部署方法的部分指令。换言之，一个或多个计算设备 150 的组合可以共同执行用于执行应用部署方法的指令。

本申请实施例还提供了一种包含指令的计算机程序产品。所述计算机程序产品可以是包含指令的，能够运行在计算设备上或被储存在任何可用介质中的软件或程序产品。当所述计算机程序产品在至少一个计算设备上运行时，使得至少一个计算设备执行应用部署方法。

本申请实施例还提供了一种计算机可读存储介质。所述计算机可读存储介质可以是计算设备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的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）等。该计算机可读存储介质包括指令，所述指令指示计算设备执行应用部署方法，或指示计算设备执行应用部署方法。

最后应说明的是：以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的保护范围。

权 利 要 求 书

- 1.一种应用部署方法，其特征在于，所述方法包括：
- 提供组件编排界面，所述组件编排界面包括：组件选择区和组件编排区，所述组件选择区包括：同容器组件和多个预设组件，所述同容器组件用于指示与其自身关联的所述多个预设组件运行在同一容器中，所述组件编排区用于为用户提供编排组件的操作区，所述多个预设组件包括业务组件和/或逻辑组件；
- 响应于用户在所述组件编排区的操作，在所述组件编排区中创建第一同容器组件，并确定与所述第一同容器组件关联的多个第一预设组件，多个预设组件包含所述多个第一预设组件；
- 响应于接收到的应用部署请求，部署包含所述多个第一预设组件的目标应用，其中，所述多个第一预设组件运行在所述第一同容器组件对应的第一容器中。
- 2.根据权利要求1所述的方法，其特征在于，在所述接收应用部署请求前，所述方法还包括：
- 响应于接收到的应用发布请求，根据所述组件编排区中的全部组件，生成所述目标应用，所述全部组件包括所述第一同容器组件和所述多个第一预设组件。
- 3.根据权利要求1或2所述的方法，其特征在于，所述组件编排界面还包括：组件配置区；所述方法还包括：
- 从所述第一同容器组件对应的组件配置区中获取目标资源量的大小，所述目标资源量为所述第一同容器组件对应的所述第一容器所需的资源量；
- 为所述第一容器分配所述目标资源量。
- 4.根据权利要求1-3中任一项所述的方法，其特征在于，所述根据所述组件编排区中的全部组件，生成所述目标应用，包括：
- 解压每个所述第一预设组件，得到每个所述第一预设组件对应的解压文件；
- 将所述多个第一预设组件对应的全部解压文件进行压缩，得到融合组件；
- 根据所述融合组件，生成所述目标应用。
- 5.根据权利要求4所述的方法，其特征在于，在得到所述融合组件之后，所述方法还包括：
- 将调用的每个所述第一预设组件的接口均更新为所述融合组件的接口。
- 6.根据权利要求1-5中任一项所述的方法，其特征在于，根据所述组件编排区中的全部组件，生成所述目标应用，包括：
- 根据所述组件编排区中的全部组件和所述全部组件中各个组件之间的连接关系，生成所述目标应用；其中，所述全部组件包括：所述多个第一预设组件，所述全部组件中各个组件之间的连接关系包括：所述第一预设组件之间的连接关系。
- 7.根据权利要求6所述的方法，其特征在于，所述全部组件还包括：至少一个第二预设组件，所述第二预设组件为所述多个预设组件中任一组件；所述全部组件中各个组件之间的连接关系还包括：所述第二预设组件与边缘组件之间的连接关系，所述边缘组件为所述多个第一预设组件中与所述第二预设组件存在连接关系的组件；所述至少一个第二预设组件包括：至少一个第三预设组件，所述边缘组件包括：边缘输入组件，所述第三预设组件为向所述边缘输入组件传递参数的组件；所述根据所述组件编排区中的全部组件，生成所述目标应用，还包括：
- 获取每个第三预设组件的输出参数；
- 将每个所述第三预设组件的输出参数进行聚合，得到所述第一同容器组件的输入参数集合；
- 根据每个所述第三预设组件与所述边缘输入组件之间的连接关系，建立所述输入参数集合中各个参数与所述边缘输入组件的输入参数之间的映射关系。
- 8.根据权利要求7所述的方法，其特征在于，所述至少一个第二预设组件还包括：至少一个第四预设组件，所述边缘组件还包括：边缘输出组件，所述边缘输出组件为向所述第四预设组件传递参数的组件；所述根据所述组件编排区中的全部组件，生成所述目标应用，还包括：
- 获取每个所述边缘输出组件的输出参数；
- 将每个所述边缘输出组件的输出参数进行聚合，得到所述第一同容器组件的输出参数集合；
- 根据每个所述第四预设组件与所述边缘输出组件之间的连接关系，建立所述输出参数集合中各个参数与所述第四预设组件的输入参数之间的映射关系。

9.根据权利要求 7 或 8 所述的方法,其特征在于,部署后的所述目标应用还包括:每个所述第二预设组件对应的第二容器,所述第二容器用于运行对应的所述第二预设组件。

10.根据权利要求 1-9 中任一项所述的方法,其特征在于,在运行所述第一容器的情况下,多个所述第一预设组件之间通过目标数据结构传递参数,所述目标数据结构为在内存中使用的数据结构。

11.一种应用部署装置,其特征在于,所述装置包括:

界面提供模块,用于提供组件编排界面,所述组件编排界面包括:组件选择区和组件编排区,所述组件选择区包括:同容器组件和多个预设组件,所述同容器组件用于指示与其自身关联的所述多个预设组件运行在同一容器中,所述组件编排区用于为用户提供编排组件的操作区,所述多个预设组件包括业务组件和/或逻辑组件;

处理模块,用于响应于用户在所述组件编排区的操作,在所述组件编排区中创建第一同容器组件,并确定与所述第一同容器组件关联的多个第一预设组件,多个预设组件包含所述多个第一预设组件;

所述处理模块,还用于响应于接收到的应用部署请求,部署包含所述多个第一预设组件的目标应用,其中,所述多个第一预设组件运行在所述第一同容器组件对应的第一容器中。

12.根据权利要求 11 所述的装置,其特征在于,

所述处理模块,还用于响应于接收到的应用发布请求,根据所述组件编排区中的全部组件,生成所述目标应用,所述全部组件包括所述第一同容器组件和所述多个第一预设组件。

13.根据权利要求 11 或 12 所述的装置,其特征在于,所述组件编排界面还包括:组件配置区;所述装置还包括:获取模块;

所述获取模块,用于从所述第一同容器组件对应的组件配置区中获取目标资源量的大小,所述目标资源量为所述第一同容器组件对应的所述第一容器所需的资源量;

为所述第一容器分配所述目标资源量。

14.根据权利要求 11-13 中任一项所述的装置,其特征在于,

所述处理模块,还用于解压每个所述第一预设组件,得到每个所述第一预设组件对应的解压文件;

所述处理模块,还用于将所述多个第一预设组件对应的全部解压文件进行压缩,得到融合组件;

所述处理模块,还用于根据所述融合组件,生成所述目标应用。

15.根据权利要求 14 所述的装置,其特征在于,

所述处理模块,还用于将调用的每个所述第一预设组件的接口均更新为所述融合组件的接口。

16.根据权利要求 11-15 中任一项所述的装置,其特征在于,

所述处理模块,还用于根据所述组件编排区中的全部组件和所述全部组件中各个组件之间的连接关系,生成所述目标应用;其中,所述全部组件包括:所述多个第一预设组件,所述全部组件中各个组件之间的连接关系包括:所述第一预设组件之间的连接关系。

17.根据权利要求 16 所述的装置,其特征在于,所述全部组件还包括:至少一个第二预设组件,所述第二预设组件为所述多个预设组件中任一组件;所述全部组件中各个组件之间的连接关系还包括:所述第二预设组件与边缘组件之间的连接关系,所述边缘组件为所述多个第一预设组件中与所述第二预设组件存在连接关系的组件;所述至少一个第二预设组件包括:至少一个第三预设组件,所述边缘组件包括:边缘输入组件,所述第三预设组件为向所述边缘输入组件传递参数的组件;

所述处理模块,还用于获取每个第三预设组件的输出参数;

所述处理模块,还用于将每个所述第三预设组件的输出参数进行聚合,得到所述第一同容器组件的输入参数集合;

所述处理模块,还用于根据每个所述第三预设组件与所述边缘输入组件之间的连接关系,建立所述输入参数集合中各个参数与所述边缘输入组件的输入参数之间的映射关系。

18.根据权利要求 17 所述的装置,其特征在于,所述至少一个第二预设组件还包括:至少一

个第四预设组件，所述边缘组件还包括：边缘输出组件，所述边缘输出组件为向所述第四预设组件传递参数的组件；

所述处理模块，还用于获取每个所述边缘输出组件的输出参数；

5 所述处理模块，还用于将每个所述边缘输出组件的输出参数进行聚合，得到所述第一同容器组件的输出参数集合；

所述处理模块，还用于根据每个所述第四预设组件与所述边缘输出组件之间的连接关系，建立所述输出参数集合中各个参数与所述第四预设组件的输入参数之间的映射关系。

19.根据权利要求 17 或 18 所述的装置，其特征在于，部署后的所述目标应用还包括：每个所述第二预设组件对应的第二容器，所述第二容器用于运行对应的所述第二预设组件。

10 20.根据权利要求 11-19 中任一项所述的装置，其特征在于，在运行所述第一容器的情况下，多个所述第一预设组件之间通过目标数据结构传递参数，所述目标数据结构为在内存中使用的数据结构。

21.一种计算设备集群，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；

15 所述至少一个计算设备的处理器用于执行所述至少一个计算设备的存储器中存储的指令，由所述计算设备集群执行如权利要求 1-10 中任一项所述的方法。

22.一种包含指令的计算机程序产品，其特征在于，当所述指令被电子设备运行时，由所述电子设备执行如权利要求 1-10 中任一项所述的方法。

20 23.一种计算机可读存储介质，其特征在于，包括计算机程序指令，当所述计算机程序指令被计算设备集群运行时，由所述计算设备集群执行如权利要求 1-10 中任一项所述的方法。