



(12) 发明专利

(10) 授权公告号 CN 113641456 B

(45) 授权公告日 2023.06.13

(21) 申请号 202110948981.2

(22) 申请日 2021.08.18

(65) 同一申请的已公布的文献号
申请公布号 CN 113641456 A

(43) 申请公布日 2021.11.12

(73) 专利权人 中国联合网络通信集团有限公司
地址 100033 北京市西城区金融大街21号

(72) 发明人 李铭轩 魏进武 张呈宇 博格利
张基恒

(74) 专利代理机构 北京天昊联合知识产权代理
有限公司 11112
专利代理师 彭瑞欣 姜春咸

(51) Int. Cl.
G06F 9/455 (2006.01)
G06F 8/60 (2018.01)
G06F 9/50 (2006.01)

(56) 对比文件
CN 108462746 A, 2018.08.28

WO 2017045424 A1, 2017.03.23

CN 107577538 A, 2018.01.12

CN 111796838 A, 2020.10.20

CN 112199178 A, 2021.01.08

CN 113157379 A, 2021.07.23

IN 202047012700 A, 2020.05.15

US 2011231442 A1, 2011.09.22

谷瑞军. 基于Docker容器的轻量级大数据实验平台构建研究.《金陵科技学院学报》.2019, 第35卷(第4期), 第1-5页.

Nannan Zhao等. Large-Scale Analysis of the Docker Hub Dataset.《2019 IEEE International Conference on Cluster Computing (CLUSTER)》.2019, 第1-10页.

李铭轩等. 面向电信运营商的IT资源微服务化方案.《信息通信技术》.2017, 第11卷(第2期), 第48-55页.

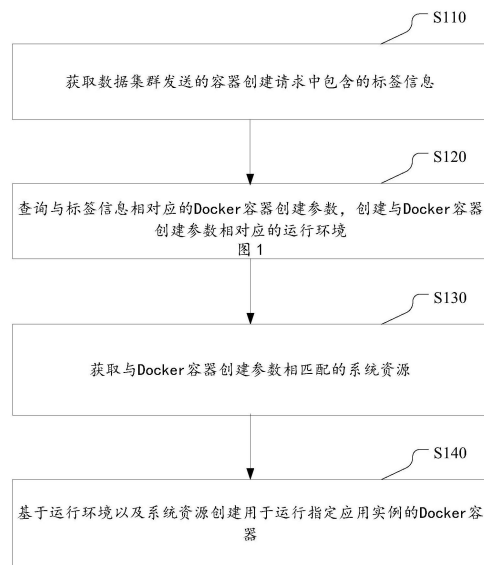
审查员 汪平

权利要求书2页 说明书11页 附图6页

(54) 发明名称
数据集群的部署方法、装置及系统

(57) 摘要

本发明公开了一种数据集群的部署方法、装置及系统,该方法包括:获取数据集群发送的容器创建请求中包含的标签信息;查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境;获取与所述Docker容器创建参数相匹配的系统资源;基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器。该方式能够通过自动创建Docker容器的方式实现集群的自动化部署,提升了集群部署效率。



1. 一种数据集群的部署方法,其特征在于,包括:
 - 获取数据集群发送的容器创建请求中包含的标签信息;
 - 查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境;
 - 获取与所述Docker容器创建参数相匹配的系统资源;
 - 基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器;
 - 其中,所述获取数据集群发送的容器创建请求中包含的标签信息之前,进一步包括:通过集群管理框架提供的接口,申请与所述指定应用实例相对应的虚拟资源;
 - 则所述查询与所述标签信息相对应的Docker容器创建参数包括:
 - 查询注册表中存储的与所述标签信息相对应的Docker容器创建参数;其中,所述标签信息与所述Docker容器创建参数之间的对应关系基于所述集群管理框架确定;
 - 其中,所述查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境包括:
 - 若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集群的参数,则创建基于所述集群管理框架的Docker运行环境;
 - 若所述与所述标签信息相对应的Docker容器创建参数为对应于第二类数据集群的参数,则创建基于所述集群管理框架以及YARN框架的Docker运行环境;
 - 其中,所述若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集群的参数,则创建基于所述集群管理框架的Docker运行环境包括:
 - 根据所述对应于第一类数据集群的参数,确定所述第一类数据集群对应的框架种类;
 - 若所述第一类数据集群的框架种类为第一量级类框架,则创建基于所述集群管理框架的第一Docker运行环境;
 - 若所述第一类数据集群的框架种类为第二量级类框架,则创建基于所述集群管理框架的第二Docker运行环境;
 - 其中,所述第一量级类框架所支持的数据规模小于所述第二量级类框架所支持的数据规模,且所述第一Docker运行环境中的Docker容器型号小于所述第二Docker运行环境中的Docker容器型号。
2. 根据权利要求1所述的方法,其特征在于,所述集群管理框架为mesos框架,且所述系统资源进一步包括:服务器资源、文件资源、以及网络资源。
3. 根据权利要求1-2任一所述的方法,其特征在于,所述获取数据集群发送的容器创建请求中包含的标签信息包括:通过Kubernetes的监听机制,监听数据集群发送的容器创建请求;
 - 并且,所述基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器之后,进一步包括:针对空闲的Docker容器执行销毁操作。
4. 一种数据集群的部署装置,其特征在于,包括:
 - 标签获取模块,适于获取数据集群发送的容器创建请求中包含的标签信息;
 - 查询模块,适于查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境;
 - 资源获取模块,适于获取与所述Docker容器创建参数相匹配的系统资源;

容器创建模块,适于基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器;

其中,所述获取数据集发送的容器创建请求中包含的标签信息之前,进一步包括:通过集群管理框架提供的接口,申请与所述指定应用实例相对应的虚拟资源;

则所述查询与所述标签信息相对应的Docker容器创建参数包括:

查询注册表中存储的与所述标签信息相对应的Docker容器创建参数;其中,所述标签信息与所述Docker容器创建参数之间的对应关系基于所述集群管理框架确定;

其中,所述查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境包括:

若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集的参数,则创建基于所述集群管理框架的Docker运行环境;

若所述与所述标签信息相对应的Docker容器创建参数为对应于第二类数据集的参数,则创建基于所述集群管理框架以及YARN框架的Docker运行环境;

其中,所述若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集的参数,则创建基于所述集群管理框架的Docker运行环境包括:

根据所述对应于第一类数据集的参数,确定所述第一类数据集对应的框架种类;

若所述第一类数据集的框架种类为第一量级类框架,则创建基于所述集群管理框架的第一Docker运行环境;

若所述第一类数据集的框架种类为第二量级类框架,则创建基于所述集群管理框架的第二Docker运行环境;

其中,所述第一量级类框架所支持的数据规模小于所述第二量级类框架所支持的数据规模,且所述第一Docker运行环境中的Docker容器型号小于所述第二Docker运行环境中的Docker容器型号。

5. 一种数据集的部署系统,其特征在于,包括:权利要求4所述的数据集的部署装置、数据集、以及集群管理框架。

6. 一种电子设备,其特征在于,包括:

一个或多个处理器;

存储装置,其上存储有一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现根据权利要求1-3任意一项所述的方法;

一个或多个I/O接口,连接在所述处理器与存储器之间,配置为实现所述处理器与存储器的信息交互。

7. 一种计算机可读介质,其上存储有计算机程序,所述程序被处理器执行时实现根据权利要求1-3任意一项所述的方法。

数据集群的部署方法、装置及系统

技术领域

[0001] 本发明涉及通信技术,尤其涉及一种数据集群的部署方法、装置及系统。

背景技术

[0002] 随着大数据技术的不断发展,各项业务对于大数据的需求也越来越强烈,随着业务的不断发展,原有的大数据存储技术已无法满足数据量日益增长的业务需求。相应的,现有的平台技术架构已经无法满足数据的存储需求以及数据处理能力的需求。为了解决上述问题,提出了一种基于Share-Disk架构实现的大数据解决方案。Share-Disk架构通过多个服务器节点共享一个存储来实现大数据集群。

[0003] 但是,发明人在实现本发明的过程中发现,上述架构至少存在以下缺陷:由于同一时间集群中只有一台服务器工作,其他服务器只能作为冗余服务器,从而导致硬件资源严重浪费;并且,在存储方面容易出现单点故障。而且,该架构中的集群需要人工部署,操作繁琐。

发明内容

[0004] 鉴于上述问题,本发明提供了一种数据集群的部署方法、装置及系统,以解决现有技术中的集群需要人工部署,操作繁琐的问题

[0005] 根据本发明的一个方面,提供了一种数据集群的部署方法,包括:

[0006] 获取数据集群发送的容器创建请求中包含的标签信息;

[0007] 查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境;

[0008] 获取与所述Docker容器创建参数相匹配的系统资源;

[0009] 基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器。

[0010] 可选地,所述获取数据集群发送的容器创建请求中包含的标签信息之前,进一步包括:通过集群管理框架提供的接口,申请与所述指定应用实例相对应的虚拟资源;

[0011] 则所述查询与所述标签信息相对应的Docker容器创建参数包括:

[0012] 查询注册表中存储的与所述标签信息相对应的Docker容器创建参数;其中,所述标签信息与所述Docker容器创建参数之间的对应关系基于所述集群管理框架确定。

[0013] 可选地,所述集群管理框架为mesos框架,且所述系统资源进一步包括:服务器资源、文件资源、以及网络资源。

[0014] 可选地,所述查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境包括:

[0015] 若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集群的参数,则创建基于所述集群管理框架的Docker运行环境;

[0016] 若所述与所述标签信息相对应的Docker容器创建参数为对应于第二类数据集群的参数,则创建基于所述集群管理框架以及YARN框架的Docker运行环境。

[0017] 可选地,所述若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集的参数,则创建基于所述集群管理框架的Docker运行环境包括:

[0018] 根据所述对应于第一类数据集的参数,确定所述第一类数据集对应的框架种类;

[0019] 若所述第一类数据集的框架种类为第一量级类框架,则创建基于所述集群管理框架的第一Docker运行环境;

[0020] 若所述第一类数据集的框架种类为第二量级类框架,则创建基于所述集群管理框架的第二Docker运行环境;

[0021] 其中,所述第一量级类框架所支持的数据规模小于所述第二量级类框架所支持的数据规模,且所述第一Docker运行环境中的Docker容器型号小于所述第二Docker运行环境中的Docker容器型号。

[0022] 可选地,所述获取数据集发送的容器创建请求中包含的标签信息包括:通过Kubernetes的监听机制,监听数据集发送的容器创建请求;

[0023] 并且,所述基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器之后,进一步包括:针对空闲的Docker容器执行销毁操作。

[0024] 根据本发明的又一个方面,提供了一种数据集的部署装置,包括:

[0025] 标签获取模块,适于获取数据集发送的容器创建请求中包含的标签信息;

[0026] 查询模块,适于查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境;

[0027] 资源获取模块,适于获取与所述Docker容器创建参数相匹配的系统资源;

[0028] 容器创建模块,适于基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器。

[0029] 可选地,标签获取模块进一步适于:通过集群管理框架提供的接口,申请与所述指定应用实例相对应的虚拟资源;则查询模块具体适于:查询注册表中存储的与所述标签信息相对应的Docker容器创建参数;其中,所述标签信息与所述Docker容器创建参数之间的对应关系基于所述集群管理框架确定。

[0030] 可选地,所述集群管理框架为mesos框架,且所述系统资源进一步包括:服务器资源、文件资源、以及网络资源。

[0031] 可选地,所述查询模块具体适于:若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集的参数,则创建基于所述集群管理框架的Docker运行环境;若所述与所述标签信息相对应的Docker容器创建参数为对应于第二类数据集的参数,则创建基于所述集群管理框架以及YARN框架的Docker运行环境。

[0032] 可选地,所述若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集的参数,则查询模块具体适于:

[0033] 根据所述对应于第一类数据集的参数,判断所述第一类数据集对应的框架种类;

[0034] 若所述第一类数据集的框架种类为第一量级类框架,则创建基于所述集群管理框架的第一Docker运行环境;

[0035] 若所述第一类数据集的框架种类为第二量级类框架,则创建基于所述集群管理

框架的第二Docker运行环境；

[0036] 其中,所述第一量级类框架所支持的数据规模小于所述第二量级类框架所支持的数据规模,且所述第一Docker运行环境中的Docker容器型号小于所述第二Docker运行环境中的Docker容器型号。

[0037] 可选地,标签获取模块具体适于:通过Kubernetes的监听机制,监听数据集群发送的容器创建请求;并且,容器创建模块进一步适于:针对空闲的Docker容器执行销毁操作。

[0038] 根据本发明的又一个方面,提供了一种数据集群的部署系统,包括:上述的数据集群的部署装置、数据集群、以及集群管理框架。

[0039] 根据本发明的又一个方面,提供了一种电子设备,包括:

[0040] 一个或多个处理器;

[0041] 存储装置,其上存储有一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现上述数据集群的部署方法;

[0042] 一个或多个I/O接口,连接在所述处理器与存储器之间,配置为实现所述处理器与存储器的信息交互。

[0043] 根据本发明的再一个方面,提供了一种计算机可读介质,其上存储有计算机程序,所述程序被处理器执行时实现上述数据集群的部署方法。

[0044] 在本发明提供的数据集群的部署方法、装置及系统中,获取数据集群发送的容器创建请求中包含的标签信息,查询与标签信息相对应的Docker容器创建参数,创建与Docker容器创建参数相对应的运行环境;获取与Docker容器创建参数相匹配的系统资源,基于运行环境以及系统资源创建用于运行指定应用实例的Docker容器。由此可见,该方式能够根据标签信息自动创建匹配的运行环境,进而获取系统资源以创建Docker容器。该方式基于Docker容器实现,且能够通过标签信息自动适配不同种类的数据集群,从而提升容器创建的便捷性和可靠性。在该方式中,同一时间集群内可以有多个Docker容器同时工作,因此,硬件资源利用率较高,且能够有效避免单点故障的问题。并且,该方式能够通过自动创建Docker容器的方式实现集群的自动化部署,提升了集群部署效率。

附图说明

[0045] 图1示出了本发明实施例一提供的数据集群的部署方法的流程图;

[0046] 图2示出了本发明实施例二提供的数据集群的部署方法的流程图;

[0047] 图3示出了Share-Disk架构示意图;

[0048] 图4示出了Share-Nothing架构示意图;

[0049] 图5示出了基于Kubernetes的大数据集群的架构示意图;

[0050] 图6示出了对应于不同的上层大数据集群的实例需求的环境配置示意图;

[0051] 图7示出了本发明又一实施例提供的数据集群的部署装置的结构图;

[0052] 图8为本发明又一实施例提供的一种电子设备的结构示意图。

具体实施方式

[0053] 为使本领域的技术人员更好地理解本发明的技术方案,下面结合附图对本发明提供的服务器进行详细描述。

[0054] 在下文中将参考附图更充分地描述示例实施例,但是所述示例实施例可以以不同形式来体现且不应被解释为限于本文阐述的实施例。反之,提供这些实施例的目的在于使本发明透彻和完整,并将使本领域技术人员充分理解本发明的范围。

[0055] 如本文所使用的,术语“和/或”包括一个或多个相关列举条目的任何和所有组合。

[0056] 本文所使用的术语仅用于描述特定实施例,且不意欲限制本发明。如本文所使用的,单数形式“一个”和“该”也意欲包括复数形式,除非上下文另外清楚指出。还将理解的是,当本说明书中使用术语“包括”和/或“由……制成”时,指定存在所述特征、整体、步骤、操作、元件和/或组件,但不排除存在或添加一个或多个其它特征、整体、步骤、操作、元件、组件和/或其群组。

[0057] 本文所述实施例可借助本发明的理想示意图而参考平面图和/或截面图进行描述。因此,可根据制造技术和/或容限来修改示例图示。因此,实施例不限于附图中所示的实施例,而是包括基于制造工艺而形成的配置的修改。因此,附图中例示的区具有示意性属性,并且图中所示区的形状例示了元件的区的具体形状,但并不旨在是限制性的。

[0058] 除非另外限定,否则本文所用的所有术语(包括技术和科学术语)的含义与本领域普通技术人员通常理解的含义相同。还将理解,诸如那些在常用字典中限定的那些术语应当被解释为具有与其在相关技术以及本发明的背景下的含义一致的含义,且将不解释为具有理想化或过度形式上的含义,除非本文明确如此限定。

[0059] 实施例一、

[0060] 图1示出了本发明实施例一提供的数据集群的部署方法的流程图。如图1所示,该方法包括以下步骤:

[0061] 步骤S110:获取数据集群发送的容器创建请求中包含的标签信息。

[0062] 其中,数据集群通常为大数据集群,既可以为基于MySQL的数据集群,也可以为基于Hadoop的数据集群,本发明对数据集群的种类不做限定。容器创建请求用于创建容器,以运行指定应用实例。其中,容器创建请求中包含标签信息,该标签信息用于指示数据集群的种类、容器的类型等各类与容器创建过程相关的信息内容。本发明不限定标签信息的具体内涵,只要能够对容器创建过程提供参考即可。

[0063] 步骤S120:查询与标签信息相对应的Docker容器创建参数,创建与Docker容器创建参数相对应的运行环境。

[0064] 具体的,通过查询注册表的方式,获取与标签信息相对应的Docker容器创建参数,进而创建与Docker容器创建参数相对应的运行环境。其中,在注册表中,存储了各种标签信息与Docker容器创建参数之间的对应关系。相应的,通过查询注册表,能够获取到与标签信息相对应的Docker容器创建参数,进而创建与Docker容器创建参数相对应的运行环境。

[0065] 步骤S130:获取与Docker容器创建参数相匹配的系统资源。

[0066] 具体的,根据Docker容器创建参数,确定创建对应的Docker容器时所需的系统资源的种类和数量,并基于系统资源的种类和数量,获取对应的系统资源。具体获取时,向对应的资源管理服务器发送资源申请请求,从而根据资源管理服务器反馈的资源申请结果,得到对应的系统资源。

[0067] 步骤S140:基于运行环境以及系统资源创建用于运行指定应用实例的Docker容器。

[0068] 具体的,基于上述步骤中创建的运行环境以及获取到的系统资源,创建用于运行指定应用实例的Docker容器。其中,该Docker容器依赖于上述的运行环境,且在运行过程中需要使用上述的系统资源。

[0069] 由此可见,该方式能够根据标签信息自动创建匹配的运行环境,进而获取系统资源以创建Docker容器。该方式基于Docker容器实现,且能够通过标签信息自动适配不同种类的数据集群,从而提升容器创建的便捷性和可靠性。在该方式中,同一时间集群内可以有多个Docker容器同时工作,因此,硬件资源利用率较高,且能够有效避免单点故障的问题。并且,该方式能够通过自动创建Docker容器的方式实现集群的自动化部署,提升了集群部署效率。

[0070] 实施例二、

[0071] 图2示出了本发明实施例二提供的数据集群的部署方法的流程图。如图2所示,该方法包括以下步骤:

[0072] 步骤S200:通过集群管理框架提供的接口,申请与指定应用实例相对应的虚拟资源。

[0073] 其中,集群管理框架可以为mesos框架,用于提供底层虚拟资源。相应的,本实施例中的执行主体可以为Kubernetes集群,具体可以为Kubernetes中的节点代理,例如,可以为kubenet节点代理。通过集群管理框架mesos提供的接口,能够申请与指定应用实例相对应的虚拟资源。其中,mesos框架用于实现底层资源的调度和开销,上层架构不需要关心底层的硬件资源具体包括哪些,统一由mesos来完成底层资源的调度和管理,从而便于实现上层集群的扩展。

[0074] 步骤S210:获取数据集群发送的容器创建请求中包含的标签信息。

[0075] 其中,数据集群通常为大数据集群,既可以为基于MySQL的数据集群,也可以为基于Hadoop的数据集群,本发明对数据集群的种类不做限定。容器创建请求用于创建容器,以运行指定应用实例。其中,容器创建请求中包含标签信息,该标签信息用于指示数据集群的种类、容器的类型等各类与容器创建过程相关的信息内容。本发明不限定标签信息的具体内涵,只要能够对容器创建过程提供参考即可。

[0076] 另外,为了便于实现集群的灵活部署,在获取数据集群发送的容器创建请求中包含的标签信息时,具体通过Kubernetes的监听机制,监听数据集群发送的容器创建请求。

[0077] 步骤S220:查询与标签信息相对应的Docker容器创建参数,创建与Docker容器创建参数相对应的运行环境。

[0078] 具体的,通过查询注册表的方式,获取与标签信息相对应的Docker容器创建参数,进而创建与Docker容器创建参数相对应的运行环境。其中,在注册表中,存储了各种标签信息与Docker容器创建参数之间的对应关系。具体实施时,查询注册表中存储的与标签信息相对应的Docker容器创建参数;其中,标签信息与Docker容器创建参数之间的对应关系基于集群管理框架确定。相应的,通过查询注册表,能够获取到与标签信息相对应的Docker容器创建参数,进而创建与Docker容器创建参数相对应的运行环境。

[0079] 其中,创建与Docker容器创建参数相对应的运行环境时,通过以下方式实现:

[0080] 若与标签信息相对应的Docker容器创建参数为对应于第一类数据集群的参数,则创建基于集群管理框架的Docker运行环境。其中,第一类数据集群是指:mesos框架能够直

接支持的数据集群。第一类数据集群包括:MySQL数据集群、Oracle数据集群和MPP(大规模并行处理,Massively Parallel Processing)数据集群。例如,当Docker容器创建参数为对应于MySQL数据集群的参数时,说明数据集群为MySQL类型,此时,由于mesos框架对MySQL类型的支持程度较好,因此,可以直接创建基于集群管理框架的Docker运行环境,而无需借助其他框架的辅助。

[0081] 若与标签信息相对应的Docker容器创建参数为对应于第二类数据集群的参数,则创建基于集群管理框架以及YARN框架的Docker运行环境。其中,第二类数据集群是指:mesos框架无法直接支持的数据集群。第二类数据集群包括:Hadoop数据集群。例如,当Docker容器创建参数为对应于Hadoop数据集群的参数时,说明数据集群为Hadoop类型,此时,由于mesos框架对Hadoop类型的支持程度较低,因此,需要创建YARN(Yet Another Resource Negotiator,另一种资源协调者)框架进行辅助,以借助YARN框架以及mesos框架的组合共同实现。

[0082] 另外,当与标签信息相对应的Docker容器创建参数为对应于第一类数据集群的参数时,进一步根据对应于第一类数据集群的参数,判断第一类数据集群对应的框架种类,并基于框架种类执行不同的处理过程:若第一类数据集群的框架种类为第一量级类框架,则创建基于集群管理框架的第一Docker运行环境;若第一类数据集群的框架种类为第二量级类框架,则创建基于集群管理框架的第二Docker运行环境;其中,第一量级类框架所支持的数据规模小于第二量级类框架所支持的数据规模,且第一Docker运行环境中的Docker容器型号小于第二Docker运行环境中的Docker容器型号。其中,第一量级类框架是指:量级较轻、数据规模较小的框架,如MySQL数据集群的框架;第二量级类框架是指:量级较重、数据规模较大的框架,如Oracle数据集群和MPP数据集群的框架。由此可见,在本申请中,能够针对数据集群的框架种类,设置不同型号的Docker容器。

[0083] 步骤S230:获取与Docker容器创建参数相匹配的系统资源。

[0084] 具体的,根据Docker容器创建参数,确定创建对应的Docker容器时所需的系统资源的种类和数量,并基于系统资源的种类和数量,获取对应的系统资源。具体获取时,向对应的资源管理服务器发送资源申请请求,从而根据资源管理服务器反馈的资源申请结果,得到对应的系统资源。其中,系统资源进一步包括:服务器资源、文件资源、以及网络资源。

[0085] 步骤S240:基于运行环境以及系统资源创建用于运行指定应用实例的Docker容器。

[0086] 具体的,基于上述步骤中创建的运行环境以及获取到的系统资源,创建用于运行指定应用实例的Docker容器。其中,该Docker容器依赖于上述的运行环境,且在运行过程中需要使用上述的系统资源。

[0087] 另外,在基于运行环境以及系统资源创建用于运行指定应用实例的Docker容器之后,进一步针对空闲的Docker容器执行销毁操作,以实现资源的自动回收。

[0088] 为了便于理解,下面以一个具体示例为例,详细描述本实施例中的实现方式。

[0089] 首先,针对Kubernetes集群进行介绍:Kubernetes是一种容器集群管理系统,其提供应用部署、维护、扩展机制等功能,利用Kubernetes能方便地管理跨机器运行容器化的应用,其主要功能包括如下几个方面:

[0090] 使用Dockers对应用程序包装(package)、实例化(instantiate)、运行(run);

[0091] 以集群的方式运行、管理跨机器的容器；

[0092] 解决Docker跨机器容器之间的通讯问题；

[0093] Kubernetes的自我修复机制使得容器集群总是运行在用户期望的状态。

[0094] 当前Kubernetes支持GCE、vSphere、CoreOS、OpenShift、Azure等平台，除此之外，也可以直接运行在物理机上。Kubernetes主要包括：Kubecfg、主应用程序接口服务器(Master API server)、Kublet(节点代理)、Minion(主机)以及Proxy(代理)。

[0095] 然后，介绍几种可选的大数据集群创建机制。随着大数据的不断发展，现有的数据量相比较于过去有了几何级的提升，因此大数据的存储和维护不能仅仅依靠单一的数据库服务器解决，而是通过大数据集群的方式来实现分布式的数据存储和共享。目前主流的大数据集群创建机制主要包括以下几种形式：

[0096] (1) Share-Disk架构

[0097] Share-Disk架构是通过多个服务器节点共享一个存储来实现大数据集群，其最简单的Share-Disk架构如图3所示，包括数据库引擎A、数据库引擎B以及共享存储。在此基础上，Share-Disk架构又分为单活和双活，双活即为集群中的每一个节点都可以同时对外提供服务，而单活为集群中只有一个节点可对外提供服务，集群中的其他服务器作为冗余在“活”的节点出现故障时接替该服务器成为对外提供服务的节点。上述架构的弊端至少包括：硬件资源的严重浪费，同一时间集群中只有一台服务器工作，其他服务器只能作为冗余服务器；集群无法提升性能，因为只有一台服务器可用；存储方面存在单点故障，除非在存储层级保证高可用，通常需要昂贵的存储区域网络(Storage Area Network, 简称SAN)存储；架构仅仅可以做到服务器层面的高可用，无法带来性能的提升，也无法解决存储单点故障的问题。

[0098] (2) Share-Nothing架构

[0099] Share-Nothing架构又分为两种，首先是分布式架构。将数据库中的数据按照某一标准分布到多台机器中，查询或插入时按照条件查询或插入对应的分区。

[0100] 另一种是每一个节点完全独立，节点之间通过网络连接，通常是通过光纤等专用网络。其架构如图4所示：包括数据库引擎A、数据库引擎B、独立存储1以及独立存储2。其中，在Share-Nothing架构中，每一个节点都拥有自己的内存和存储，都保留数据的完整副本。通常来说，又可以分为两种，可以负载均衡和不可以负载均衡。对于不可以负载均衡的集群，在不可负载均衡的技术中，集群中的节点被分为主节点和辅助节点，主节点对外提供服务，辅助节点作为热备或暖备，同时有可能使得辅助节点提供只读服务。该架构的优势在于：辅助节点数据和主节点保持同步或准同步，当搭配第三方仲裁后，可以实现自动的故障转移，从而实现高可用；辅助节点由于和主节点完全独立且数据同步或准同步，因此主节点出现数据损坏后，可以从辅助节点恢复数据；由于Share-Nothing架构使用了本地存储(或SAN)，相较于Share-Disk架构在慢速网络时有非常大的性能优势。当然，该架构的弊端至少包括：因为辅助节点无法对外提供服务或只能提供只读服务，因此该类集群的扩展能力非常有限；对性能没有提升，因为涉及到各节点的数据同步，甚至带来性能的下降；辅助节点如果可读，虽然提升性能，但需要修改前端应用程序，对应用程序不透明。

[0101] 由此可见，对于上述几种大数据集群方案而言，在高可用性、可扩展性或性能方面存在一些问题，因而无法满足目前的大数据应用的业务需求。虽然上述技术方案能够解决

大数据存储和集群分布存储、查询等问题,但是还存在如下问题:

[0102] 首先,硬件资源严重浪费:上述几种技术架构为了实现共享存储的能力,往往是以牺牲硬件资源作为代价的,诸如现有技术的Share-Disk方案,在集群运作方式上,往往是只提供一台服务器作为运行服务,而其他服务器只作为冗余服务,因此硬件资源造成严重浪费;其次为了保障单节点的稳定性,往往采用昂贵的存储区域网络(Storage Area Network,简称SAN)进行存储,因此组网成本较高。

[0103] 其次,上述技术框架为了保证主数据节点和辅助数据节点的同步,辅助节点无法对外提供服务或者只能提供只读服务,为的是保证辅助节点的数据安全性,虽然保证了数据的安全性,但是同时也造成了集群的扩展能力非常有限,对于性能提升没有任何帮助,同时也涉及各节点的数据同步,其性能也有所下降。

[0104] 再次,存储空间消耗加大:现有的技术架构为了保证数据集的完整性,因此对于每一个节点上的内存数据集都要保证一份完整的数据集,随着节点数量的增加,那么同样的数据集在不同的数据节点上需要保存多份,从而造成存储空间的浪费。

[0105] 为了解决上述问题,本申请采用基于Kubernetes架构的方式,通过Docker来管理数据集完成数据节点的存储和保存,并且在开销方面通过Kubernetes架构对于Docker的管理,来完成数据节点的开销和生命周期的管理,从而降低存储开销。其中,基于Kubernetes的大数据集识别系统按照Kubernetes的架构,将集群封装成Docker来统一管理和进行调度。利用Kubernetes的管理能力和负载均衡能力来实现大数据集群的存储和处理能力。具体的,本申请中的方案主要涉及如下几部分:硬件资源管理、Docker封装、Docker管理及调度以及集群扩展机制等。

[0106] 首先,介绍本申请中的方案的系统架构。图5示出了基于Kubernetes的大数据集的架构示意图。如图5所示,该系统架构从下至上依次包括:虚拟资源管理、mesos框架、Kubernetes、Docker层、大数据集群、Ku8以及数据应用层。依据上述架构图,底层硬件资源通过虚拟化的方式形成虚拟资源管理,统一由mesos来实现底层资源的调度和开销,上层架构不需要关心底层的硬件资源的种类和数量,统一由mesos来完成底层资源的调度和管理,从而为上层集群的扩展提供了便利。其次,基于Kubernetes框架完成对Docker的管理,大数据集群主要封装在Docker中,再通过Kubernetes中的Ku8管理来进行管理和运行。从而实现对于大数据集群的动态扩展和自动识别,并且将新的大数据集注册和加入到整个基于Kubernetes框架的大数据集管理系统中。

[0107] 接下来,介绍面向大数据集群的Docker封装机制。基于Docker的封装机制是在底层虚拟资源被封装完成以后,统一由mesos对上层Docker提供资源的接口,通过Docker根据应用实例申请底层资源,根据上层大数据集群的实例需求自动配置环境,并且统一打包形成镜像以供上层大数据集群运行。由于目前的虚拟资源的框架mesos对于上层大数据集群框架的支持不同,需要在通过Docker封装时配置不同的运行环境,如:MySQL数据库对底层mesos支持较好,可以直接运行在Docker中;Hadoop数据库对于底层mesos支持较弱,需要通过YARN框架与底层mesos进行交互。图6示出了对应于不同的上层大数据集群的实例需求的环境配置示意图。如图6所示,该环境配置至少包括:调度层、数据库框架层、Docker层以及资源管理层。依据上述的封装机制,面向现有的大数据集群框架对于底层资源管理层的适配不同,主要采用如下三种Docker封装机制:

[0108] (1) 面向小型的、轻应用的数据库框架(如MySQL数据库框架),可以实现Docker的直接封装,直接通过底层资源管理层进行调度和管理;

[0109] (2) 面向Hadoop集群的大数据框架,目前新版本的Hadoop框架基本上是基于YARN框架来进行运行和配置的,因此在封装成Docker时需要同时配置YARN框架,再通过YARN框架调度底层资源管理层;

[0110] (3) 面向重型的、大规模的大数据框架(如Oracle和MPP等框架),可以通过重型Docker容器进行封装,在物理资源分配方面,可以集中更多的物理资源来配置Docker,从而保证面向Oracle和MPP等重型数据仓库的运行开销资源。

[0111] 由此可见,根据数据集群对应的数据框架的种类的不同,可将Docker封装方式分为多种。其中,当数据集群对应的数据框架为Hadoop集群时,在封装成Docker时需要同时配置YARN框架。另外,还可以根据数据集群对应的数据框架的量级和规模,分别封装小型Docker和重型Docker。

[0112] 最后,介绍面向大数据集群的Docker的封装识别流程:基于上述面向不同类型的大数据集群框架的不同所面临的Docker封装机制不同,为了做到自动识别大数据集群框架的类型并且启动特定的Docker封装机制来进行资源环境的配置,从而减少人工干预的过程。本示例中的Docker封装识别流程包括以下步骤:

[0113] 步骤一、面向大数据集群通过标签(Label)向Kublet发送容器创建请求;

[0114] 步骤二、Kublet将请求中的标签发送给注册表,通过注册表查询根据大数据集群创建Docker的配置需求;

[0115] 步骤三、注册表中维系着创建不同类型的大数据集群的Docker创建参数,并将对应的参数返回给Kublet;

[0116] 步骤四、Kublet根据参数配置向服务器(Server)请求资源;

[0117] 步骤五、Kublet根据参数配置向文件设备(file)申请文件资源;

[0118] 步骤六、Kublet根据参数配置向网络设备(Http)申请网络资源等;

[0119] 步骤七、Kublet根据申请的资源创建和管理Pod。

[0120] 根据上述的流程,通过Kubernetes中的Kublet自动创建容器来配置面向大数据集群的容器,以运行框架。

[0121] 另外,再介绍一下本示例中的集群扩展机制以及容器销毁机制。

[0122] 其中,集群的扩展主要通过Docker的封装机制来实现。利用Kubernetes的监听机制实现新集群的接入,通过Kubernetes的Kubecfg发起Docker创建请求,从而能够实现基于Docker的大数据集群环境的创建。并且,通过Kubernetes的通信机制完成数据之间的同步和通信能力。另外,为了保证底层资源的高利用率,对于大数据集群的资源减少和重复利用建立容器销毁机制。对于下架的大数据集群,通过Kubernetes框架的管理机制对空出来的Docker及时销毁,以实现资源的自动收回。

[0123] 综上可知,本示例提供的基于Kubernetes的大数据集群自动识别的方法,能够将现有技术中通过人工配置、建立部署的大数据集群,转换成基于Docker封装的、自动识别的部署方式,从而显著提升了自动化部署的能力,提升了集群的部署效率和灵活性。

[0124] 图7示出了本发明又一实施例提供的一种数据集群的部署装置,包括:

[0125] 标签获取模块81,适于获取数据集群发送的容器创建请求中包含的标签信息;

[0126] 查询模块82,适于查询与所述标签信息相对应的Docker容器创建参数,创建与所述Docker容器创建参数相对应的运行环境;

[0127] 资源获取模块83,适于获取与所述Docker容器创建参数相匹配的系统资源;

[0128] 容器创建模块84,适于基于所述运行环境以及所述系统资源创建用于运行指定应用实例的Docker容器。

[0129] 可选地,标签获取模块进一步适于:通过集群管理框架提供的接口,申请与所述指定应用实例相对应的虚拟资源;则查询模块具体适于:查询注册表中存储的与所述标签信息相对应的Docker容器创建参数;其中,所述标签信息与所述Docker容器创建参数之间的对应关系基于所述集群管理框架确定。

[0130] 可选地,所述集群管理框架为mesos框架,且所述系统资源进一步包括:服务器资源、文件资源、以及网络资源。

[0131] 可选地,所述查询模块具体适于:若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集的参数,则创建基于所述集群管理框架的Docker运行环境;若所述与所述标签信息相对应的Docker容器创建参数为对应于第二类数据集的参数,则创建基于所述集群管理框架以及YARN框架的Docker运行环境。

[0132] 可选地,所述若所述与所述标签信息相对应的Docker容器创建参数为对应于第一类数据集的参数,则查询模块具体适于:

[0133] 根据所述对应于第一类数据集的参数,判断所述第一类数据集对应的框架种类;

[0134] 若所述第一类数据集的框架种类为第一量级类框架,则创建基于所述集群管理框架的第一Docker运行环境;

[0135] 若所述第一类数据集的框架种类为第二量级类框架,则创建基于所述集群管理框架的第二Docker运行环境;

[0136] 其中,所述第一量级类框架所支持的数据规模小于所述第二量级类框架所支持的数据规模,且所述第一Docker运行环境中的Docker容器型号小于所述第二Docker运行环境中的Docker容器型号。

[0137] 可选地,标签获取模块具体适于:通过Kubernetes的监听机制,监听数据集发送的容器创建请求;并且,容器创建模块进一步适于:针对空闲的Docker容器执行销毁操作。

[0138] 本发明又一实施例还提供了一种数据集的部署系统,包括:上述的数据集的部署装置、数据集、以及集群管理框架。

[0139] 另外,参照图8,本发明又一实施例提供了一种电子设备,其包括:

[0140] 一个或多个处理器501;

[0141] 存储器502,其上存储有一个或多个程序,当一个或多个程序被一个或多个处理器执行,使得一个或多个处理器实现上述任意一项的数据集的部署方法;

[0142] 一个或多个I/O接口503,连接在处理器与存储器之间,配置为实现处理器与存储器的信息交互。

[0143] 其中,处理器501为具有数据处理能力的器件,其包括但不限于中央处理器(CPU)等;存储器502为具有数据存储能力的器件,其包括但不限于随机存取存储器(RAM,更具体如SDRAM、DDR等)、只读存储器(ROM)、带电可擦可编程只读存储器(EEPROM)、闪存(FLASH);

I/O接口(读写接口)503连接在处理器501与存储器502间,能实现处理器501与存储器502的信息交互,其包括但不限于数据总线(Bus)等。

[0144] 在一些实施例中,处理器501、存储器502和I/O接口503通过总线相互连接,进而与计算设备的其它组件连接。

[0145] 最后,本发明又一实施例提供了一种计算机可读介质,其上存储有计算机程序,程序被处理器执行时实现上述任意一种数据集群的部署方法。

[0146] 本领域普通技术人员可以理解,上文中所发明方法中的全部或某些步骤、系统、装置中的功能模块/单元可以被实施为软件、固件、硬件及其适当的组合。在硬件实施方式中,在以上描述中提及的功能模块/单元之间的划分不一定对应于物理组件的划分;例如,一个物理组件可以具有多个功能,或者一个功能或步骤可以由若干物理组件合作执行。某些物理组件或所有物理组件可以被实施为由处理器,如中央处理器、数字信号处理器或微处理器执行的软件,或者被实施为硬件,或者被实施为集成电路,如专用集成电路。这样的软件可以分布在计算机可读介质上,计算机可读介质可以包括计算机存储介质(或非暂时性介质)和通信介质(或暂时性介质)。如本领域普通技术人员公知的,术语计算机存储介质包括在用于存储信息(诸如计算机可读指令、数据结构、程序模块或其它数据)的任何方法或技术中实施的易失性和非易失性、可移除和不可移除介质。计算机存储介质包括但不限于RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘(DVD)或其它光盘存储、磁盒、磁带、磁盘存储或其它磁存储装置、或者可以用于存储期望的信息并且可以被计算机访问的任何其它的介质。此外,本领域普通技术人员公知的是,通信介质通常包含计算机可读指令、数据结构、程序模块或者诸如载波或其它传输机制之类的调制数据信号中的其它数据,并且可包括任何信息递送介质。

[0147] 本文已经发明了示例实施例,并且虽然采用了具体术语,但它们仅用于并仅应当被解释为一般说明性含义,并且不用于限制的目的。在一些实例中,对本领域技术人员显而易见的是,除非另外明确指出,否则可单独使用与特定实施例相结合描述的特征、特性和/或元素,或可与其它实施例相结合描述的特征、特性和/或元件组合使用。因此,本领域技术人员将理解,在不脱离由所附的权利要求阐明的本发明的范围的情况下,可进行各种形式和细节上的改变。

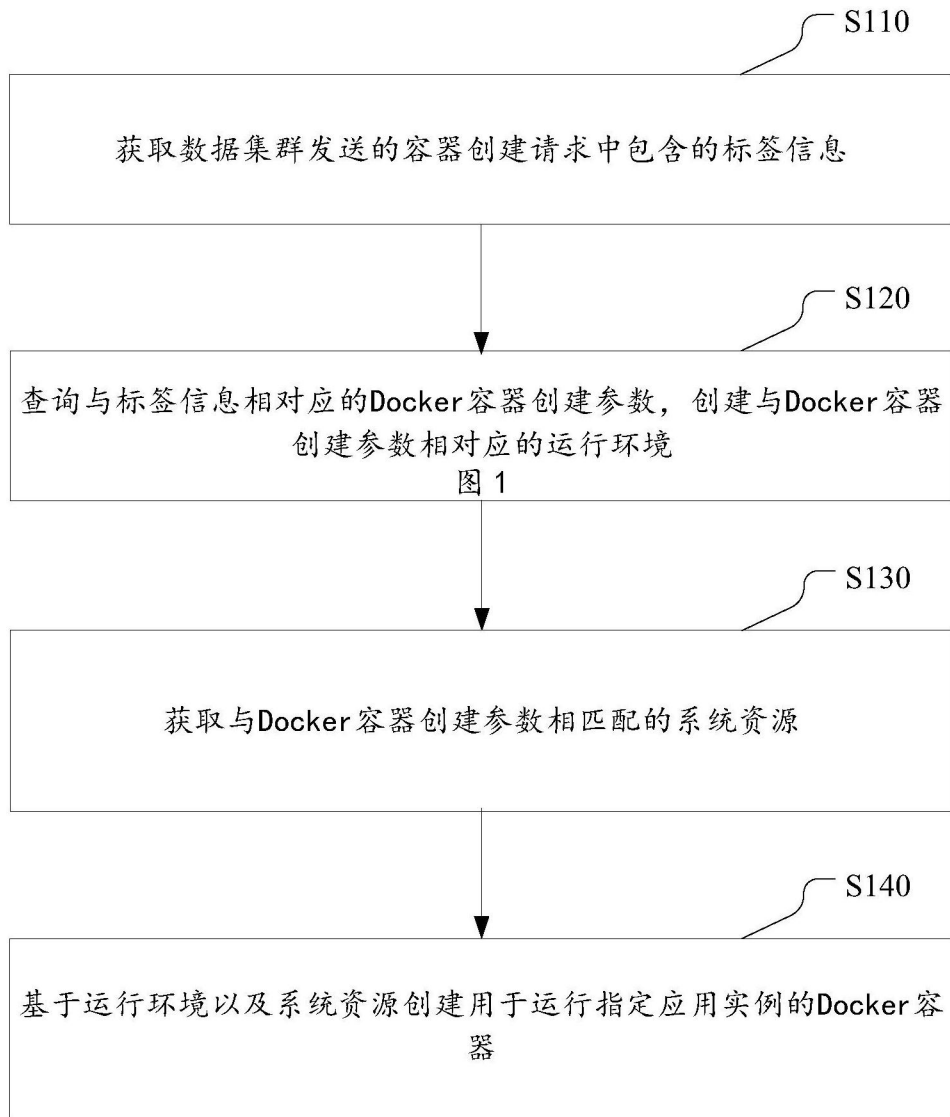


图1

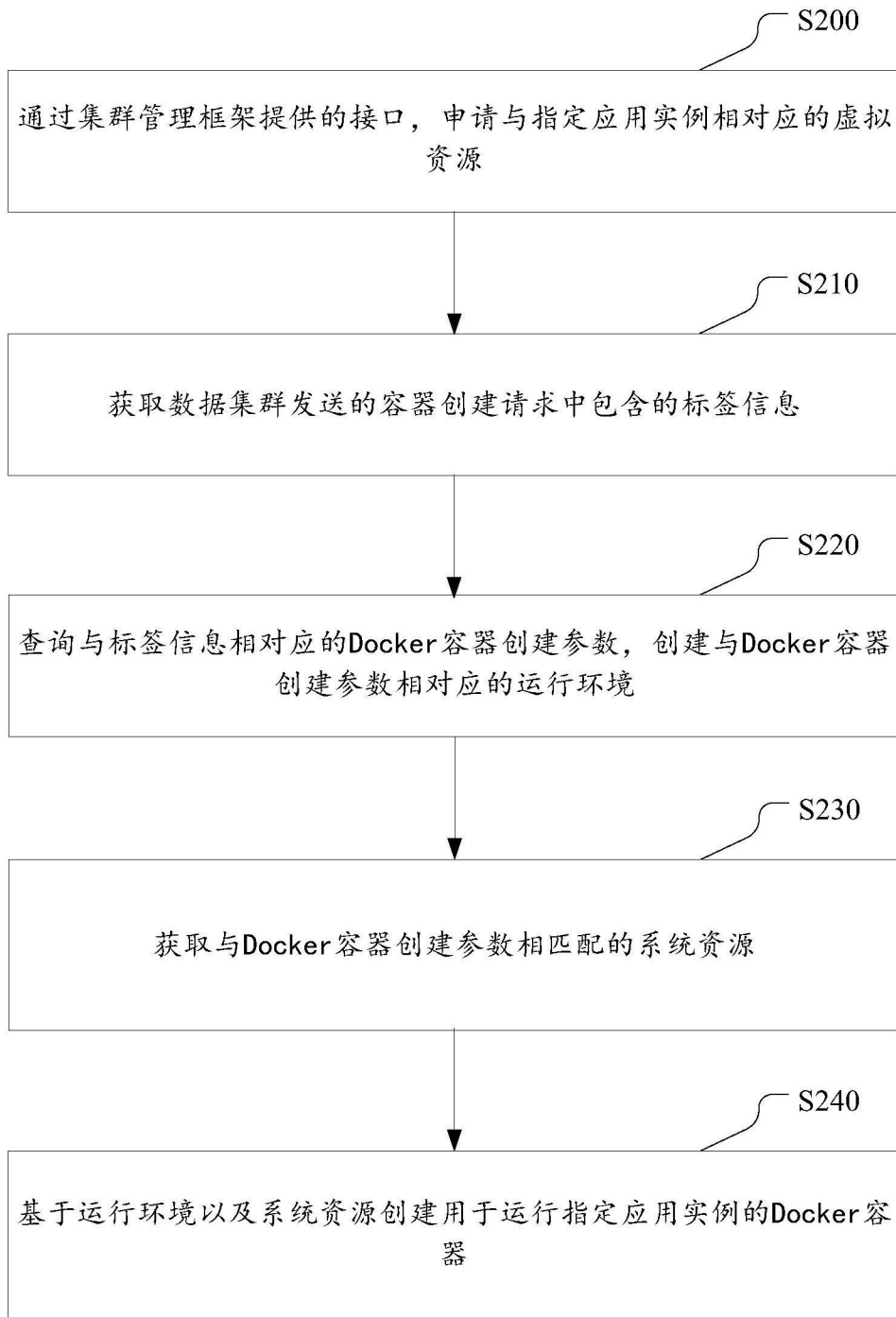


图2

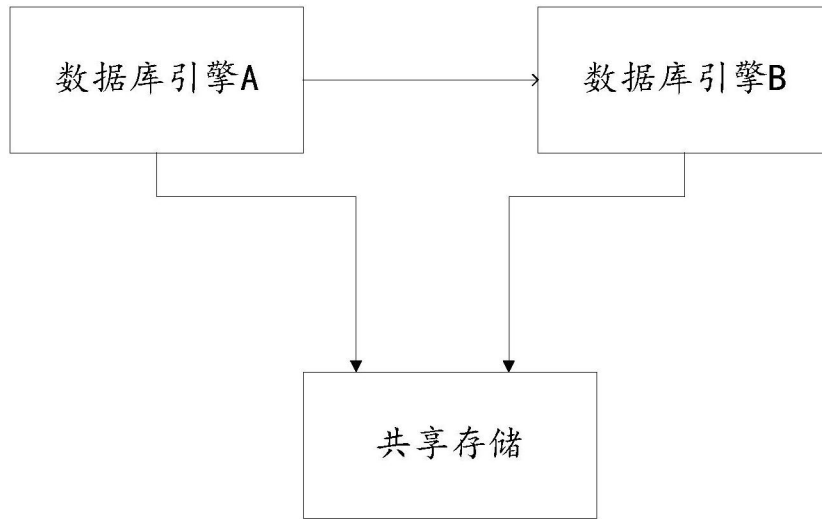


图3

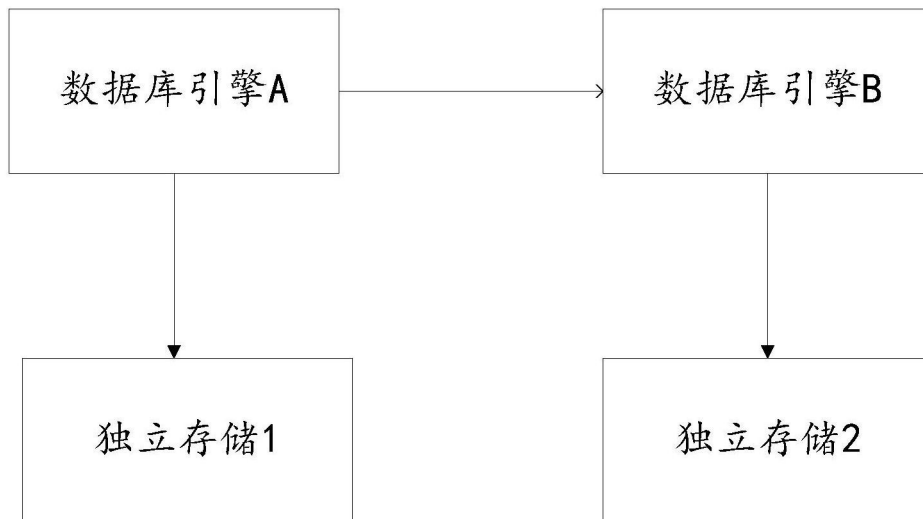


图4

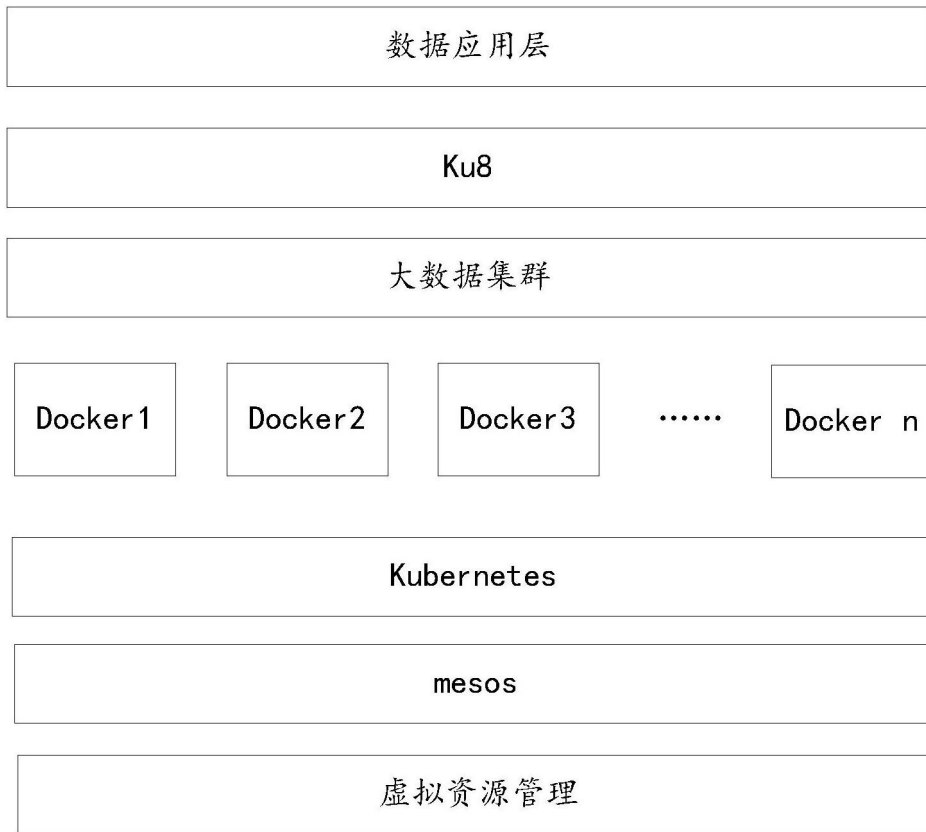


图5

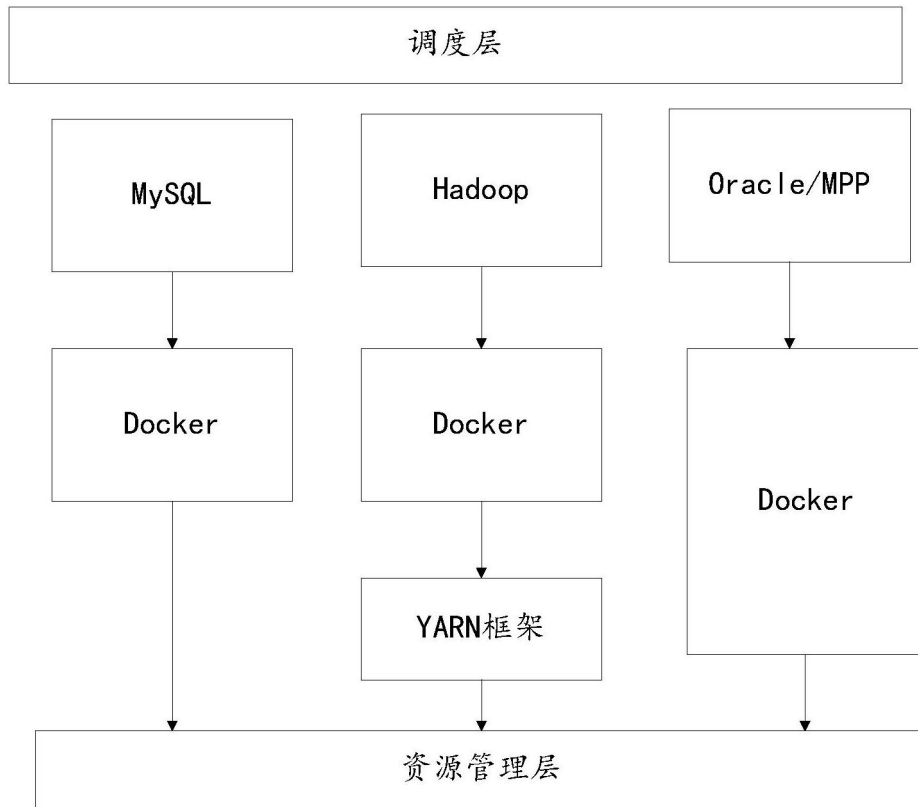


图6

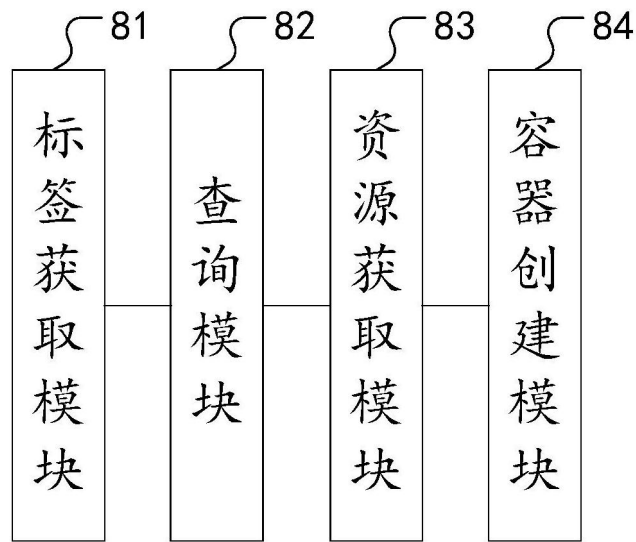


图7

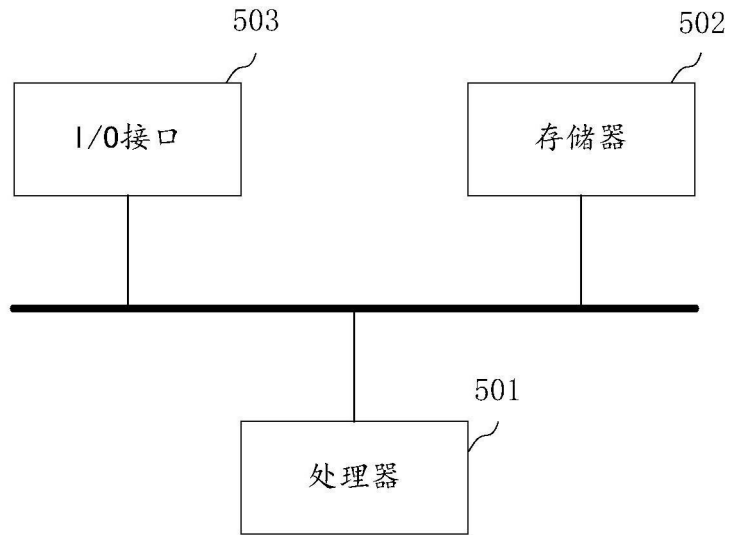


图8