(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 108279967 A
(43)申请公布日 2018.07.13

(21)申请号 201711016482.X

(22)申请日 2017.10.25

(71)申请人 国云科技股份有限公司
地址 523808 广东省东莞市松山湖科技产业园区松科苑14号楼

(72)发明人 刘勇彬 季统凯

(74)专利代理机构 北京科亿知识产权代理事务所(普通合伙) 11350
代理人 汤东凤

(51)Int.Cl.
G06F 9/455(2006.01)

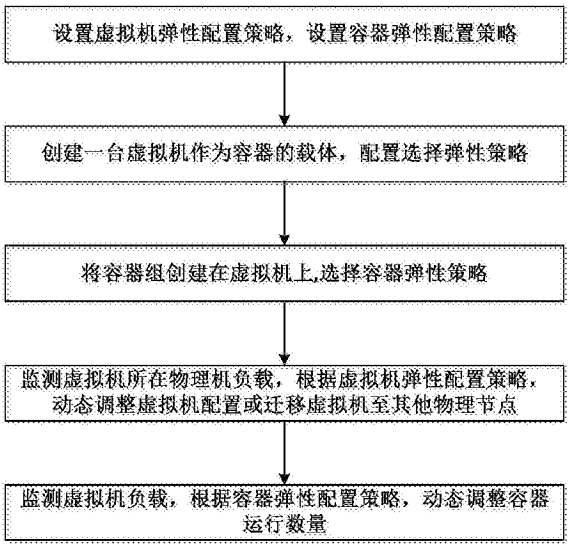
权利要求书2页 说明书4页 附图1页

(54)发明名称

一种虚拟机与容器混合调度方法

(57)摘要

本发明涉及云计算技术领域,特别是指一种虚拟机与容器混合调度方法。本发明所述方法是将容器作为虚拟机的内置应用创建在虚拟机上,同时,赋予容器和虚拟机弹性配置策略;根据弹性配置策略创建容器和虚拟机的配置或迁移。本发明方法将容器创建在虚拟机上,作为虚拟机的内置应用,即保障了容器与虚拟机之间的安全问题,又结合了虚拟机的弹性调度策略,提高了物理资源利用率;本发明方法可提高容器可用性,利用虚拟机的弹性调度避免了容器在迁移过程的停机问题。



CN 108279967 A

1.一种虚拟机与容器混合调度方法，其特征在于：所述方法是将容器作为虚拟机的内置应用创建在虚拟机上，同时，赋予容器和虚拟机弹性配置策略；根据弹性配置策略创建容器和虚拟机的配置或迁移。

2.根据权利要求1所述的方法，其特征在于：所述的方法具体包括如下步骤：

步骤1：设置虚拟机弹性配置策略，设置容器弹性配置策略；

步骤2：创建一台虚拟机作为容器的载体，配置选择弹性策略；

步骤3：将容器组创建在虚拟机上，选择容器弹性策略；

步骤4：监测虚拟机所在物理机负载，根据虚拟机弹性配置策略，动态调整虚拟机配置或迁移虚拟机至其他物理节点；

步骤5：监测虚拟机负载，根据容器弹性配置策略，动态调整容器运行数量。

3.根据权利要求2所述的方法，其特征在于：

所述的容器，容器为应用程序提供了隔离的运行空间：每个容器内都包含一个独享的完整用户环境空间，并且一个容器内的变动不会影响其他容器的运行环境；

所述的容器的载体，即容器创建在虚拟机所在操作系统上，与其他虚拟机等同，容器只是作为虚拟机的特殊应用来对待；

所述的容器组，是指多个容器的集合，可以单个应用集群容器，也可以多个，创建的时候可以限制容器组里面容器最大最小个数。

4.根据权利要求2所述的方法，其特征在于：

所述的设置虚拟机弹性配置策略，是指根据所在物理服务器CPU、内存的利用率调整虚拟机CPU、内存大小的策略；

在创建虚拟机时，设置虚拟机最小CPU、最小内存及最大CPU、最大内存；当物理机CPU利用率低于30％时，虚拟机CPU大小加1，数据可配置；物理机内存利用率大于80％时，虚拟机内存大小减2G，数据可配置；

当遇到降低虚拟机配置时，可以优先配置迁移虚拟机方式；即当前虚拟机所在物理服务器负载较高需要降低虚拟机配置时，可以查看是否其他物理服务器有足够的资源提供现有虚拟机使用，若有则可以采用迁移再监测方式。

5.根据权利要求3所述的方法，其特征在于：

所述的设置虚拟机弹性配置策略，是指根据所在物理服务器CPU、内存的利用率调整虚拟机CPU、内存大小的策略；

在创建虚拟机时，设置虚拟机最小CPU、最小内存及最大CPU、最大内存；当物理机CPU利用率低于30％时，虚拟机CPU大小加1，数据可配置；物理机内存利用率大于80％时，虚拟机内存大小减2G，数据可配置；

当遇到降低虚拟机配置时，可以优先配置迁移虚拟机方式；即当前虚拟机所在物理服务器负载较高需要降低虚拟机配置时，可以查看是否其他物理服务器有足够的资源提供现有虚拟机使用，若有则可以采用迁移再监测方式。

6.根据权利要求1至5任一项所述的方法，其特征在于：

所述的容器弹性配置策略，主要是根据虚拟机的配置情况而决定，在创建容器组的时候可以设置容器组最小及最大数量，当前运行数量；当虚拟机配置增加时，适当增加容器数量，当虚拟机配置降低时，适当减少容器数量；

如果是计算密集型应用，对于CPU的要求就比较高；当虚拟机CPU增加后，增加容器的数值就应该适当减小，避免造成虚拟机配置过载的情况；

所述的计算密集型应用，即程序系统大部分在做计算、逻辑判断、循环导致CPU占用率很高的应用。

一种虚拟机与容器混合调度方法

技术领域

[0001] 本发明涉及云计算技术领域，特别是指一种虚拟机与容器混合调度方法。

背景技术

[0002] 随着云计算技术的发展，以Docker为代表的容器技术一度被认为是虚拟化技术的替代品，然而这两种技术之间并不是不可调和的。随着容器技术的逐渐成熟，容器和虚拟机并存已经成为一种必然。由于容器本身隔离性问题，很多公有云厂家目前是将容器创建在虚拟机里面的，用于隔绝对其他虚拟机造成影响。也有一些厂家将容器和虚拟机同时创建在同一台物理服务器上以此来提高物理资源的利用率，这些方式存在以下弊端：

[0003] (1)比较呆板，公有云将容器创建在虚拟机上的方式，缺乏调度，只是单纯的为用户提供容器服务，无法有效提高物理资源利用率。

[0004] (2)安全性差，容器和虚拟机同时创建在同一台物理服务器必然会相互影响，特别是容器的隔离性较差，会导致与虚拟机资源争抢的情况。

[0005] 为此，需要一种调度策略，让用户既能安全的同时使用虚拟机与容器，又能提高物理资源利用率的解决方案。

发明内容

[0006] 本发明解决的技术问题在于提供一种虚拟机与容器混合调度方法，解决传统方法存在的不足，为用户提供一种既能安全的同时使用虚拟机与容器，又能提高物理资源利用率的解决方案。

[0007] 本发明解决上述技术问题的技术方案是：

[0008] 所述方法是将容器作为虚拟机的内置应用创建在虚拟机上，同时，赋予容器和虚拟机弹性配置策略；根据弹性配置策略创建容器和虚拟机的配置或迁移。

[0009] 所述的方法具体包括如下步骤：

[0010] 步骤1：设置虚拟机弹性配置策略，设置容器弹性配置策略；

[0011] 步骤2：创建一台虚拟机作为容器的载体，配置选择弹性策略；

[0012] 步骤3：将容器组创建在虚拟机上，选择容器弹性策略；

[0013] 步骤4：监测虚拟机所在物理机负载，根据虚拟机弹性配置策略，动态调整虚拟机配置或迁移虚拟机至其他物理节点；

[0014] 步骤5：监测虚拟机负载，根据容器弹性配置策略，动态调整容器运行数量。

[0015] 所述的容器，容器为应用程序提供了隔离的运行空间：每个容器内都包含一个独享的完整用户环境空间，并且一个容器内的变动不会影响其他容器的运行环境；为了能达到这种效果，容器技术使用了一系列的系统级别的机制诸如利用Linux namespaces来进行空间隔离，通过文件系统的挂载点来决定容器可以访问哪些文件，通过cgroups来确定每个容器可以利用多少资源。此外容器之间共享同一个系统内核，这样当同一个库被多个容器使用时，内存的使用效率会得到提升。

[0016] 所述的容器的载体，即容器创建在虚拟机所在操作系统上，与其他虚拟机等同，容器只是作为虚拟机的特殊应用来对待；

[0017] 所述的容器组，是指多个容器的集合，可以单个应用集群容器，也可以多个，创建的时候可以限制容器组里面容器最大最小个数。

[0018] 所述的设置虚拟机弹性配置策略，是指根据所在物理服务器CPU、内存的利用率调整虚拟机CPU、内存大小的策略；

[0019] 在创建虚拟机时，设置虚拟机最小CPU、最小内存及最大CPU、最大内存；当物理机CPU利用率低于30％时，虚拟机CPU大小加1，数据可配置；物理机内存利用率大于80％时，虚拟机内存大小减2G，数据可配置；

[0020] 当遇到降低虚拟机配置时，可以优先配置迁移虚拟机方式；即当前虚拟机所在物理服务器负载较高需要降低虚拟机配置时，可以查看是否其他物理服务器有足够的资源提供现有虚拟机使用，若有则可以采用迁移再监测方式。

[0021] 所述的容器弹性配置策略，主要是根据虚拟机的配置情况而决定，在创建容器组的时候可以设置容器组最小及最大数量，当前运行数量；当虚拟机配置增加时，适当增加容器数量，当虚拟机配置降低时，适当减少容器数量；

[0022] 如果是计算密集型应用，对于CPU的要求就比较高；当虚拟机CPU增加后，增加容器的数值就应该适当减小，避免造成虚拟机配置过载的情况；

[0023] 所述的计算密集型应用，即程序系统大部分在做计算、逻辑判断、循环导致CPU占用率很高的应用。

[0024] 本发明方法将容器创建在虚拟机上，作为虚拟机的内置应用，即保障了容器与虚拟机之间的安全问题，又结合了虚拟机的弹性调度策略，提高了物理资源利用率；本发明方法提高容器可用性，利用虚拟机的弹性调度避免了容器在迁移过程的停机问题。

附图说明

[0025] 下面结合附图对本发明进一步说明：

[0026] 图1为本发明的流程图；

[0027] 图2为本发明实施框架图。

具体实施方式

[0028] 为使本发现的目的、技术方案和优点更加清楚，下面将结合附图并以实际实施案例作进一步详细解说，如图1、2所示，具体实施过程如下：

[0029] 1、设置虚拟机弹性配置策略，设置容器弹性配置策略；

[0030] 所述的设置虚拟机弹性配置策略，是指调整虚拟机CPU、内存大小的策略，在创建虚拟机时，设置虚拟机最小CPU、最小内存及最大CPU、最大内存，设置虚拟机弹性策略，主要根据所在物理服务器负载进行调整，比如物理机CPU利用率低于30％时，虚拟机CPU大小加1，数据可配置；比如物理机内存利用率大于80％时，虚拟机内存大小减2G，数据可配置。所有的配置策略均根据物理服务CPU、内存的利用率来进行调整，最终是为了提高物理资源利用率，当遇到降低虚拟机配置时，可以优先配置迁移虚拟机方式。

[0031] 所述的优先配置迁移虚拟机策略，即当前虚拟机所在物理服务器负载较高需要降

低虚拟机配置时，可以查看是否其他物理服务器有足够的资源提供现有虚拟机使用，若有则可以采用迁移再监测方式。

[0032] 设置参数如下：

[0033] /*

[0034] *name String 是

[0035] *type 应用类型(模板类型)String cluster(应用集群)|deploy(应用部署) 是

[0036] *description 备注 String 否

[0037] *instance.cpu 实例配置cpu个数 int

[0038] *instance.memory 实例配置内存大小(M) int

[0039] *instance.attachVolumeType 挂载云盘的类型 string Local|RBD

[0040] *instance.instanceStorageType 存储类型 int 0|1|2

[0041] *instance.isFloating 虚拟机浮动ip绑定int(0,1)

[0042] *loadbalance.isFloating 浮动ip绑定 int(0,1) 否

[0043] *loadbalance.type 负载均衡类型 string"new"

[0044] *scaling_policy.policyType 扩展方式String scaleOutFirst横向优先，scaleUpFirst纵向优先

[0045] #scaleout 横向#scaleup 纵向

[0046] *scaling_policy.minInst 最小虚拟机数 int

[0047] *scaling_policy.maxInst 最大虚拟机数 int

[0048] *scaling_policy.cpu_resize 每次扩展cpu核数 int

[0049] *scaling_policy.max_cpu 扩展cpu最大值 int

[0050] *scaling_policy.mem_resize 每次扩展内存大小(G) int

[0051] *scaling_policy.max_mem 扩展内存最大值(G) int

[0052] *alarmTypelist[n]扩展条件策略 List<String>pu_util","disk_io_read","disk_io_write"

[0053] */

[0054] 所述的容器弹性配置策略，主要是根据虚拟机的配置情况而决定，在创建容器组的时候可以设置容器组最小及最大数量，当前运行数量，当虚拟机配置增加时，适当增加容器数量，当虚拟机配置降低时，适当减少容器数量。这块需要根据容器运行的运行来决定，比如如果是计算密集型应用，对于CPU的要求就比较高，这个时候虚拟机CPU增加后，增加容器的数值就应该适当减小，避免造成虚拟机配置过载的情况。

[0055] 2、创建一台虚拟机作为容器的载体，配置选择弹性策略；

[0056] 3、将容器组创建在虚拟机上，选择容器弹性策略；

[0057] 初始化容器组内容器数量，设置容器调度策略，设置容器扩展缩减策略：

[0058]
```
private Desktop addContainer(ContainerParams cp, User loginUser) throws GCloudException {
    //添加容器组
    Container ca = new Container();
ca.setNum(cp.getNum()); //容器组内容器数
```

```
            ca.setGroup(cp.getGroup());
            ca.setLbType(cp.getLbType());//容器调度策略
            ca.setExtend(cp.getExtend());//容器扩展缩减策略
            ca.setType(cp.getType());
            ca.setUserId(loginUser.getUserId());
            ca.setStartTime(new Date());
            ca.setVolumeSize(cp.getVolumeSize());
            Long caId = iContainer Dao.saveWithIncrementId(ca);
            ca.setId(caId.intValue());
        }
        return ca;
    }
```

[0059]

[0060] 4、监测虚拟机所在物理机负载，根据虚拟机弹性配置策略，动态调整虚拟机配置或迁移虚拟机至其他物理节点；

[0061] 5、监测虚拟机负载，根据容器弹性配置策略，动态调整容器运行数量。

[0062] 调用{region_id}/monitor_resource/info.do接口获取物理负载情况

```
{
    "data":
    {
        "memory_usage": 3072,
        "storage_usage": 6144,
        "hostname": "COMPUTE_20.251.39.103",
        "core": 4,
        "storage_free": 2048,
        "memory_free": 1024,
        "storage_total": 8192,
        "memory_total": 4096,
        "cpu_util": 59.3
    },
    "success": true
}。
```

[0063]

[0064]

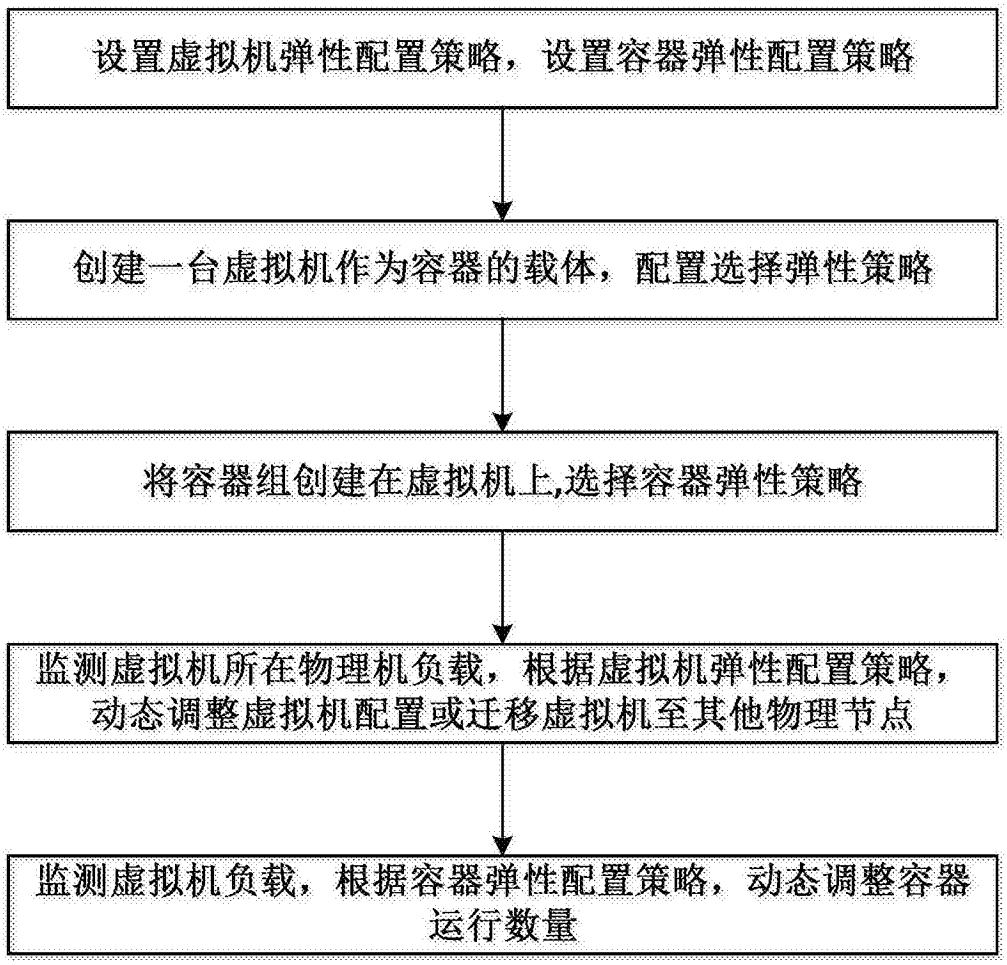

设置虚拟机弹性配置策略，设置容器弹性配置策略
创建一台虚拟机作为容器的载体，配置选择弹性策略
将容器组创建在虚拟机上,选择容器弹性策略
监测虚拟机所在物理机负载，根据虚拟机弹性配置策略，动态调整虚拟机配置或迁移虚拟机至其他物理节点
监测虚拟机负载，根据容器弹性配置策略，动态调整容器运行数量

图1

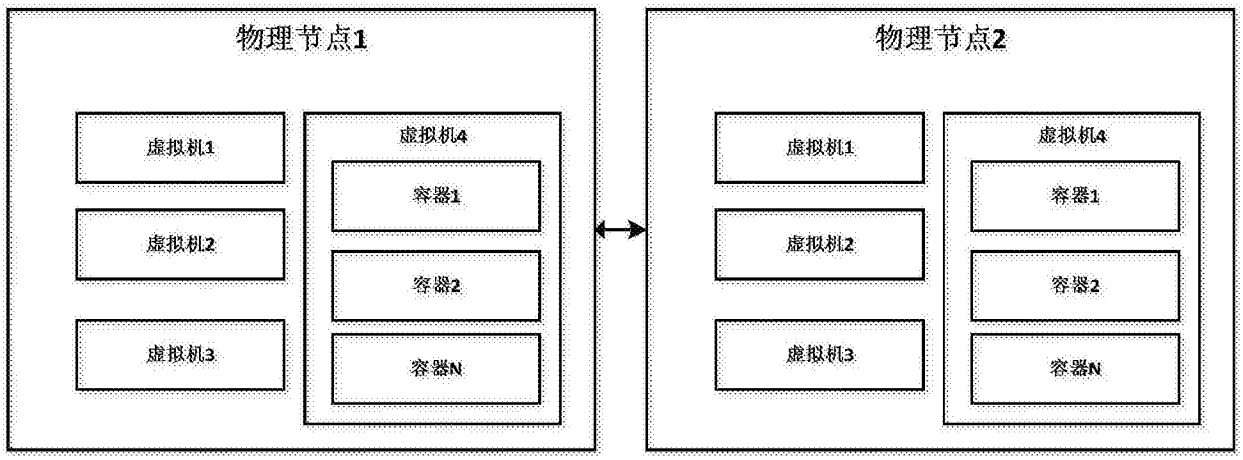

物理节点1
虚拟机1
虚拟机2
虚拟机3
虚拟机4
容器1
容器2
容器N
物理节点2
虚拟机1
虚拟机2
虚拟机3
虚拟机4
容器1
容器2
容器N

图2