基于 Ceph 的云存储应用研究

谢超群

(福建中医药大学现代教育技术中心，福建福州 350000)

摘 要：随着高校信息化的发展，高校数据中心越来越多地采用微服务架构，但传统 Docker 容器的存储技术存在着数据可靠性低、数据不能持久化、动态扩展性差等问题. 面对上述问题，本文提出一种基于 Ceph 存储集群平台和 Kubernetes 后端存储技术的高校容器云存储的解决方案，该方案利用 Ceph 构建底层云存储集群，解决了存储数据的可靠性差的问题，并利用 Kubernetes 的 Persistent Volume 和 Dynamic provisioning 存储技术实现了容器存储的持久化和动态置备，解决了数据持久化和动态扩展的问题.

关键词：Ceph；Kubernetes；Persistent Volume；数据中心

中图分类号：TP391.1 文献标识码：A 文章编号：1009 -4970(2019)02 -0043 -05

DOI:10.16594/j.cnki.41-1302/g4.2019.02.013

0 引言

随着高校数据中心微服务架构应用的不断普及，高校数据中心的 docker 容器化部署日渐增多，但 docker 容器技术存在着以下的一些问题:

(1) 有些关键容器应用的数据非常重要，但 Docker 容器技术只能默认挂载本地节点存储空间，一旦容器发生故障，存在数据丢失的风险.

(2) 很多容器应用需要持久化的保持数据，如 WEB 应用，数据库应用等，然而由于容器技术自身的局限性，当容器重启或删除之后，容器挂载的数据会丢失，无法持久保存.

(3) Docker 容器技术只能在本地划分相应的存储空间，然后挂载到容器中，无法实现划分出可以在多个服务器上共享使用的存储空间.

面对上述问题，本文利用 Ceph 分布式存储技术来搭建一个多节点高可靠性的存储集群作为容器数据的后端存储平台，同时构建一个 Kubernetes 云容器平台作为前端的管理平台，来实现容器应用数据存储的持久化和动态扩展.

1 Ceph 分布式存储简介

Ceph 是加州大学 Santa Cruz 分校的 Sage Weil 博士开发的一个开源分布式存储平台，Ceph 可以提供高可扩展性和高可靠性的存储平台，能满足对象存储、块存储和文件系统存储等多种存储类型的需求，也可以处理所有类型的数据存储，包括对象、块和文件存储类型. Ceph 的存储集群底层基于 RADOS 来开发，并且 RADOS 的高扩展性存储容量可以动态扩展到 PB 级. Ceph 还采用 CRUSH 算法动态地分布和调整数据. 该算法拥有高容错性和高一致性数据冗余机制，保证了数据的高可靠性[1]. Ceph 的体系架构图如图 1 所示.

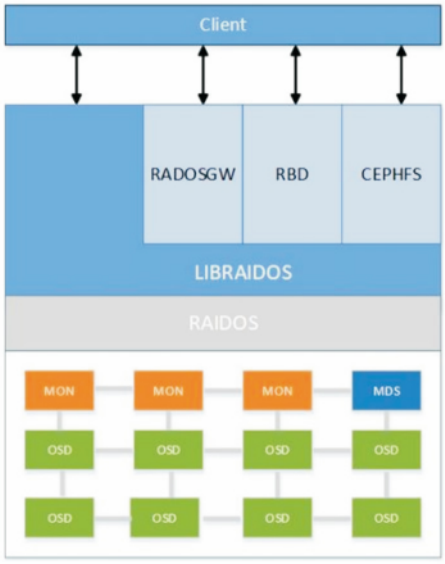

图 1 Ceph 的体系架构图

收稿日期：2018 -11 -13

基金项目：福建省教育厅项目(JAT170299)

作者简介：谢超群(1982—)，男，湖北仙桃人，硕士，工程师. 研究方向：云计算，数据中心.

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Ceph 的底层是 RADOS，它是一种稳定的、独立和完全分布式的、面向对象的存储，具备自主健康检查、自主恢复、自主管理和高级智能等特点. Ceph 中 RADOS 存储集群是所有其他客户端接口使用和部署的基础，RADOS 负责执行数据的复制、故障检测和恢复，同时利用 CRUSH 算法在集群节点之间的迁移和再平衡，确保数据的一致性和可靠性. 所有的 Ceph 存储类型包括对象存储、块存储和文件系统存储都需要从底层的 RADOS 来读写数据. RADOS 存储集群主要由 OSD 节点集群、Monitor 节点集群和 MDS 元数据节点所组成. OSD 节点与服务器的磁盘绑定，支持 xfs，btrfs，ext4 等多种文件系统. OSD 节点还负责存储对象数据，并与其他 OSD 节点通过心跳通讯共同协作完成复制数据、恢复数据、重平衡数据等功能. Monitor 节点负责监控整个 Ceph 集群，维护集群的健康状态、状态更改的历史信息以及 Ceph 集群中各种的 Cluster Map，如 OSD Map、Monitor Map、PG Map 和 CRUSH Map. Cluster Map 是 RADOS 的关键数据结构，当要存储数据时，OSD 先根据从 Monitor 节点获取的最新的 Map 图，然后通过 Object Id 和 Map 图计算出数据存储的位置. MDS 元数据节点主要负责为 Ceph 的文件系统 CephFS 提供元数据服务，MDS 元数据节点主要负责将所有的元数据存储在高速内存中，客户端对文件系统的操作将由 MDS 元数据节点快速反馈，而不用直接消耗 OSD 设备的 I/O，为 CephFS 提供了高性能保证. LIBRADOS 模块是客户端用来访问 RADOS 对象存储设备的函数库. Ceph 客户端可以用 LIBRADOS 里封装的功能和对象存储交互，所有 Ceph 存储类型都可利用 LIBRADOS 来访问底层的 OSD 存储的数据. RADOSGW 负责为客户端提供面向对象的存储服务，它是一个基于 LIBRADOS 的、向客户端提供 RESTful 接口的对象存储接口，兼容 Amazon S3 和 Open Stack Swift 的大部分 API 接口. RADOSGW 在实际应用中是一个监听 RESTfulAPI 访问的后台进程，它负责与客户端交互，同时将读写数据的操作发送到 Ceph 底层的 RADOS[2]. Ceph 的块存储服务 RBD 支持常用存储介质，如硬盘，光盘等，可以为 Ceph 客户端提供虚拟块存储服务，并挂载在服务器的文件系统内部，该块设备还支持精简制备和快照等常用存储特性. CephFS 为 Ceph 客户端提供兼容 POSIX 的分布式文件系统，通过配置 MDS 元数据管理节点，配合客户端安装对应的 Linux 内核模块的 CephFS Kernel Object 组件，来实现操作系统对 Ceph 分布式文件系统的无缝访问[3].

2 Kubernetes 后端存储技术简介

Kubernetes 根据容器 pod 的应用场景将提供的容器服务分为了三类：无状态服务、有状态服务和有状态集群服务. 无状态服务不需要保存数据，只需要对外提供服务就可以，状态数据的丢失不影响服务的正常运行；有状态服务需要保存程序运行中的状态数据，例如 Redis 服务的应用容器需保留运行中状态数据，Kubernetes 为有状态服务提供了 Volume 和 Persistent Volume 存储服务；有状态集群服务不仅需要保存状态数据，还需要对集群进行管理，同时需要动态地扩展和收缩服务所需要的存储容量. Kubernetes 采用 StorageClass 来进行 Persistent Volume 的动态存储，以达到动态伸缩集群存储容量的目的[4].

根据上述 Kubernetes 的应用场景，Kubernetes 的存储系统从低到高可以分为三个层次：Volume、Persistent Volume 和 Dynamic Provisioning. Volume 是与 Kubernetes 的容器 POD 绑定的存储，分为二种子类型：(1) emptyDir，是挂载在容器 POD 里面的匿名的空目录，当 POD 删除时 emptyDir 将被删除，主要用来 POD 之间共享状态数据；(2) hostPath，可以映射挂载宿主机中的存储路径到 POD，并独立于 POD 的生命周期，当 POD 删除时数据并不会丢失，但 POD 迁移到 Kubernetes 集群的另外一个节点时，POD 挂载的数据会丢失，因此这种存储类型只能在 POD 位于某个固定 Kubernetes 集群节点时临时存储数据[5]. Persistent Volume 是 Kubernetes 定义的独立于 POD 的存储资源对象，Persistent Volume 底层可以支持 NFS，CephFS，AWS 等各类存储. 管理员可以根据业务所需要的存储容量和访问模式来创建 Persistent Volume，POD 可以通过申明 Persistent Volume Claim 来挂载 Persistent Volume. Persistent Volume Claim 类似于 POD 的存储请求，其可以设置 Persistent Volume Claim 来静态或者动态地挂载 Persistent Volume. Persistent Volume 里存储的数据不会随着 POD 的删除而删除，同时也独立于 Kubernetes 集群的节点. Persistent Volume 是集群定义的存储，当 POD 迁移到其他的 Kubernetes 集群节点时，不影响容器 POD 继续挂载 Persistent Volume 里面的数

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

据，从而可以达到 POD 的无缝迁移[6]. Persistent Volume 的定义方式如图 2 所示:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: test
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: slow
  mountOptions:
    - hard
    - nfsvers=4.1
  nfs:
    path: /nfs/data
    server: 192.168.0.2
```

图 2　Persistent Volume 存储卷的定义

Capacity 用于定义 Persistent Volume 的大小. accessModes 定义资源的访问模式，其包含三种访问模式: (1) ReadWriteOnce，指该 Persistent Volume 能被 Kubernetes 集群里的单个节点以读写方式挂载使用; (2) ReadOnlyMany，该 Persistent Volume 能被 Kubernetes 集群里的多个节点以只读方式挂载使用; (3) ReadWriteMany，该 Persistent Volume 能被 Kubernetes 集群里的多个节点以读写方式挂载使用. Persistent Volume Reclaim Policy 用来定义 Persistent Volume 使用完成之后的处理策略，包含 Retain、Cycle、Delete 三种方式. Retain 模式是当 Persistent Volume 使用完成之后保留，等待管理员手动来处理该存储资源; Cycle 模式将会使用数据擦除命令清空 Persistent Volume 上的数据提供给新的 POD 所使用; Delete 模式删除 Persistent Volume，同时也会删除底层存储中的存储卷. Mount Options 可以指定挂载的底层存储的选项，支持 NFS，RBD，CephFS 等存储类型. Storage Class Name 用来定义设置存储资源的级别，当 Persistent Volume Claim 请求相应级别的存储时，该 Persistent Volume 将被这个 Persistent Volume Claim 所绑定.

动态存储置备是指可以根据容器 POD 的请求动态创建 Persistent Volume，不需要管理员预先创建 Persistent Volume 来满足存储资源的需求. Kubernetes 集群通过定义 Storage Class 资源对象来实现，然后用户在 Persistent Volume Claim 中添加 Storage Class 参数就可以申请动态提供存储. Storage Class 资源对象主要来描述存储类型，并提供动态存储容量供给的功能[7]. 动态存储置备的方式一般应用在有状态集群服务，需要动态伸缩存储资源的应用场景下.

3　校园容器云存储应用研究

随着高校数据中心微服务架构的增多，很多服务类似 Nginx Web 服务，Redis 缓存服务，Mysql 服务都迁移到了 Docker 容器架构上. 然而这些传统的服务迁移到 Docker 容器中遇到了很多问题，如存储数据和容器绑定只能挂载本地宿主服务器的存储，无法实现容器内数据的共享以及容器存储在服务器集群的无缝迁移. 面对传统 Docker 容器技术存在的问题，本文利用 CephFS 搭建底层存储集群，并利用 Kubernetes 后端存储技术解决上述问题.

整个校园容器云存储从上到下分为三个层次: 底层存储层，Persistent Volume 层，动态存储置备层. 底层存储层采用三台服务器搭建 Ceph FS 存储集群来实现，一台服务器作为管理节点和元数据节点安装 Ceph Monitor 和 Ceph MDS，二台服务器数据存储节点，并安装 Ceph OSD 作为存储节点. 该 Ceph 存储集群采用 CephFS 文件系统存储的方式为上层提供存储服务. CephFS 满足多个集群节点挂载可读写存储的需要，可以解决容器 POD 跨节点迁移的存储问题. 由于 Ceph 存储系统写入数据时，都会先写入到一个日志盘，当写满日志盘后才会写入到数据盘中，因此这三台服务器都配置了固态硬盘来作为日志盘，以提高整个 Ceph 存储集群的读写性能. Persistent Volume 层作为对接底层存储的服务层为 Kubernetes 集群的 Pod 提供存储服务. Persistent Volume 是 Kubernetes 集群中独立的资源对象，其可以编写 yaml 文件来生成对应的 Persistent Volume，yaml 文件的编写如图 3 所示.

该文件创建了一个 300G 的 Persistent Volume，这个 PV 采用多节点可读写的挂载方式，其可以被 Kubernetes 集群里的多个节点读写，并可以满足容器 POD 迁移到其他节点时存储无缝迁移挂载的需要，同时该 Persistent Volume 还可以被多个容器 POD 挂载来共享读写数据. Monitors 参数指定 Persistent Volume 底层对接的 Ceph 的管理节点来使用 Ceph 的底层存储服务. User 和 SecretRef 用于访问 Ceph 存储集群的用户名和访问认证密钥. Storage Class Name 指定存储的类别名称，当动态存储置备

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: cephfsdata-pv
  namespace: cephfs
spec:
  capacity:
    storage: 300Gi
  accessModes:
    - ReadWriteMany
  cephfs:
    monitors:
      - 192.168.0.1:6789
    user: admin
    secretRef:
      name: cephfs-secret
    readOnly: false
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: nginxstor
```

图 3　基于 CephFS 的 Persistent Volume 存储卷定义

存储时 Persistent Volume Claim 可以直接绑定到相应存储类型的 Persistent Volume 上.

动态存储置备层可以利用 Persistent Volume Claim 和 Storage Class 资源对象定义所需要的存储容量和存储的类别来动态申请容器 Pod 所需要的存储，其可以满足有状态集群服务对所需存储动态扩展的需求. 动态存储置备层要实现对 Ceph FS 存储的动态划分，需要在 Kubernetes 集群中部署 Cephfs Provisoner 存储动态置备的插件. Cephfs Provisoner 由 Cephfs provisoner. go 和 Cephfs provisoner. py 等脚本组成. Cephfs provisoner. go 是 Cephfs Provisoner 的核心组件，主要是监控 Kubernetes 集群中 Persistent Volume Claim 的存储资源请求事件，然后调用 Cephfs provisoner. py 的脚本创建 Persistent Volume. Cephfs provisoner. py 是用 python 语言编写的与 Cephfs 交互的命令行脚本，里面封装了对 Cephfs 的 volume 增删改查的各种操作. 安装完成 Cephfs Provisoner 后需要创建 Storage Class 和 Persistent Volume Claim 资源对象，才能实现动态存储的置备. Storage Class 和 Persistent Volume Claim 编写的 Yaml 文件如图 4 所示.

Storage Class 的 yaml 文件创建了一个名称为 Cephfs 的存储类别. namespace 参数定义命名空间为 Cephfs，该参数必须与 Persistent Volume Claim 的命名空间名称一致. provisioner 参数定义了 Cephfs 存储动态置备的支持驱动. Parameters 参数定义要连接的 Cephfs 的管理节点地址和认证账号密钥. Persistent Volume Claim 的 yaml 文件创建了名称为 claimngnix 的存储请求，namespace 参数定义了和 StorageClass 一致的命名空间. annotations 参数引用了要动态创建的存储资源类别名称 Cephfs，与之前定义好的 Storage Class 实现关联. Access Modes 参数定义的存储资源可被 Kubernetes 集群里的多个节点读写. resources 参数定义需要动态给容器 POD 配置的存储容量. 通过上述的配置动态存储置备层就可以正常运行了，该层无需手动再创建 Persistent Volume，只需要创建 PersistentVolume Claim，Cephfs Provisoner 会自动根据 Persistent Volume Claim 的请求，在后端的 Cephfs 存储集群上创建相应的 Persistent Volume.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: cephfs
  namespace: cephfs
provisioner: ceph.com/cephfs
parameters:
    monitors: 192.168.0.1:6789
    adminId: admin
    adminSecretName: ceph-secret
    adminSecretNamespace: "cephfs"

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: claimngnix
  namespace: cephfs
  annotations:
    volume.beta.Kubernetes.io/storage-class: "cephfs"
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 300Gi
```

图 4　基于 CephFS 的 StorageClass 和 PersistentVolumeClaim 的定义

5　结语

本文通过部署 Kubernetes 容器集群，配合利用 Kubernetes 后端存储组件连接 Cephfs 存储集群构建容器云存储平台，可以满足数据中心应用容器数据可靠性和持久化的要求，同时可实现容器应用集群的存储共享和动态扩展，可以解决高校数据中心传统 Docker 容器应用所面临的存储问题.

参考文献

[1] 程靓坤. 基于 Ceph 的云存储系统设计与实现 [D]. 广州: 中山大学，2014.

[2] Redhat Corporation. Ceph Documentation [EB/OL]. (2016 - 9 - 30) [2018 - 9 - 13]. http://docs.Ceph.com/docs/

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

master/radosgw/.
[3] Redhat Corporation. Ceph Documentation [EB/OL]. (2016 -9 -30) [2018 -9 -13]. http://docs.Ceph.com/docs/master/Cephfs/.
[4] 陆平,左奇. 基于 Kubernetes 的容器云平台实战 [M]. 北京:机械工业出版社. 2018:145 -146.
[5] 龚正. Kubernetes 权威指南 [M]. 北京:电子工业出版社. 2017:37 -41.
[6] The Linux Foundation. Kubernetes Documentation [EB/OL]. (2018 -5 -1) [2018 -9 -13]. https://Kubernetes.io/docs/concepts/storage/persistent -volumes/.
[7] The Linux Foundation. Kubernetes Documentation [EB/OL]. (2018 -5 -1) [2018 -9 -13]. https://Kubernetes.io/docs/concepts/storage/dynamic -provisioning/.

[责任编辑 徐 刚]

A Research on Cloud Storage Application Based on Ceph

Xie Chao-qun
(Modern Educational Technology Center, Fujian University of Traditional Chinese Medicine, Fuzhou 350000, China)

Abstract: With the development of information technology in colleges and universities, more and more micro-service architectures are applied in university data center. The storage technology of traditional docker container is faced with the problems of low data reliability, poor data persistence and poor dynamic scalability. In view of the above problems, this paper proposes a solution for university container cloud storage based on Ceph storage cluster platform and Kubernetes back-end storage technology. This solution uses Ceph to build an underlying cloud storage cluster, which solves the problem of storage data's poor reliability. Kubernetes' Persistence Volume and Dynamic Provisioning storage technology are used to realize the persistence and dynamic provisioning of container application storage, which solves the problem of data persistence and dynamic expansion.

Key words: Ceph; Kubernetes; Persistent Volume; Data Center

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net