基于 Docker 的轻量级云存储系统研究

刘毅文,黄显宁,文坤辉,米春桥
(怀化学院 计算机科学与工程学院,湖南 怀化 418000)

摘 要:在研究目前常见云存储传统系统架构的基础上,设计了一种基于 Docker 容器技术的轻量级云存储系统,并提出在虚拟化环境下采用桥接方式的网络通信方案,使容器中的虚拟网口拥有了独立的 IP 地址,在网络中的行为等同于一台虚拟服务器。轻量级的云存储系统由 2 台标准大盘位 x86 服务器和 1 台千兆交换机组成,系统内部包含元数据服务器、节点服务器的 Docker 镜像、网络通信、数据存储等模块,并考虑单点故障的场景。通过采用 Erasure code(EC)冗余机制,保证服务器的数据冗余性以及另一台宿主机中数据的完整性,为该节点继续提供完整的存储服务。通过对该系统与其他不同备份节点在磁盘利用率、写入带宽、CPU 占用率等性能点的测试和分析,结果表明,该系统可以很好地满足一般用户对低成本、高效、数据可靠的云存储需求,具有良好的性价比优势。

关键词:云存储;轻量级;Docker;数据冗余;桥接
中图分类号:TP302 **文献标识码**:A **文章编号**:1673-629X(2018)11-0159-04
doi:10.3969/j.issn.1673-629X.2018.11.035

Research on Lightweight Cloud Storage System Based on Docker

LIU Yi-wen, HUANG Xian-ning, WEN Kun-hui, MI Chun-qiao
(School of Computer Science and Engineering, Huaihua Unversity, Huaihua 418000, China)

Abstract: On the basis of studying the traditional system architecture of cloud storage, we design a lightweight cloud storage system based on Docker belonging to container technology and put forward a communication network scheme of bridging in the virtual environment, which has independent IP address in the network and is regarded as a virtual server. Cloud storage system consists of 2 sets of standard lightweight x86-servers and 1 Gigabit switch, and contains metadata server, Docker mirroring, network communication and data storage module, with considering the single point of failure scenarios. Through the use of Erasure code (EC) redundant mechanism, the integrity of data redundancy the server and another host is ensured to continue to provide full service for the node storage. The test and analysis on the disk utilization, write bandwidth and CPU utilization performance between the system and other different backup nodes system shows that the system can well satisfy the cloud storage needs from general user about low cost, high efficiency, reliable data, which has a great price advantage.

Key words: cloud storage; lightweight; Docker; data backup; bridge

0 引 言

随着计算机硬件的不断升级和云计算技术的飞速发展,云存储从云计算延伸出来,迅速发展为一种新型的网络存储技术。当云计算和处理的核心是存储和管理大量数据时,需要在云计算系统中配置大量的存储设备,然后将云计算系统转换为云存储系统。因此,云存储是一个数据存储和管理为核心的云计算系统,通过集群应用程序中网络技术或分布式文件系统等功能,使得网络中许多不同类型的存储设备通过应用软件一起工作[1],系统统一向外提供数据存储和业务访问功能。简而言之,云存储是一种新兴的解决方案,可以将存储资源放到云中供人们访问。用户可以随时随地连接到云,通过任何网络设备访问数据[2]。

随着云存储的应用越来越多,云存储产品和服务在市场上不断成熟,以满足新的数据存储需求。与此同时,云存储系统是视频云的基础,也是大数据解决方案的第一步。在将存储从硬件转换为服务后,云计算还提供了复杂的管理和调度功能[3],使云存储服务智能化,并将包含大数据分析的数据生态整合在一起,以更紧密地响应用户的各种需求。对于企业来说,要扩

收稿日期:2017-12-19 **修回日期**:2018-04-26 **网络出版时间**:2018-06-29
基金项目:湖南省自然科学基金项目(2017JJ3252);湖南省教育科学规划课题(XJK016QXX003);怀化学院科研项目(HHUY2016-04)
作者简介:刘毅文(1987-),男(侗族),硕士,讲师,研究方向为云计算、程序设计等。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20180629.1704.036.html

大云存储的市场份额,企业的快速发展可以提供机会。

文中重点讨论轻量级云存储环境部署场景,将当前标准云存储系统的最小规模(通常需要 2 个元数据服务器和 6 个数据节点)缩减到 2 个标准服务器。使用虚拟容器技术 Docker、云存储元数据节点服务器(MDS)、数据节点(DN)做 Docker 映像,部署在标准服务器上。当用户使用它时,MDS 和 DN 镜像容器直接从标准映像启动,以提供云存储服务。

1 云计算的发展过程

1.1 云计算的定义

目前,不同用户对云计算有不同的定义,即网格计算、并行计算、网络磁盘、超级服务器或像 AlphaGo 这样的高科技。然而,从学术研究人员的角度来看,云计算是一种新的商业模式,它将 IT 基础设施利用率转化为销售和服务。

1.2 云计算角色的变换

云计算在不同的历史时期扮演着不同的角色。在 20 世纪末,云计算成为一种新的概念和新技术。当时,“云计算”一词并不存在于计算机领域,然而,学者们一直在研究的网格计算[4]和并行计算实际上是云计算的早期原型。在 21 世纪初,云计算更具有代表性,但只有大型 IT 基础设施的大公司(谷歌、IBM)才能拥有。2006 年,亚马逊第一次以服务的方式出售对象存储服务,云计算从少数公司的能力发展为大众服务[5]。

1.3 云计算技术的演化过程

云计算的主要目标是应用云端的计算、存储等资源优势,突破终端的资源限制,为用户提供更加丰富的应用以及更好的用户体验。其定义一般可以概括为终端通过无线网络,以按需、易扩展的方式从云端获得所需的基础设施、平台、软件等资源或信息服务的使用与交付模式[6]。云计算技术的发展实质上是组件间的耦合越来越小,越来越孤立,资源的粒度越来越小,从专有到普通的过程。

(1)物理机时代的隔离方式—机架。

当用户需要服务时,购买物理机器。当需要更多的服务时,一个机架需要装载更多的物理机器。因此,在物理机器时代的隔离是机器级的隔离[7]。这种方式会凸现两个问题:资源粒度过粗,设备利用率不高;无法便捷地利用软件统一控制协调。

(2)操作系统的隔离方式—虚拟机。

随着虚拟化的发展,操作系统级的隔离随之出现,这通常被称为虚拟机(VM)。每一个操作系统都由许多个虚拟操作系统组成,每个操作系统都是虚拟服务器[8],同时共享硬件资源。由于单个物理机器可以创建许多虚拟机,因此资源粒度越小,利用率越高。与此同时,它带来了一个非常明显的优势,可以很容易地由软件创建、重新启动、摧毁虚拟机。当在数据中心生成大量的虚拟机,并将生成的虚拟机放在特定的管理、监视、安全和网络设施中进行辅助时,它就变成了云,即所说的 IaaS[9]。

(3)应用层面的隔离方式—PaaS。

Docker 的出现为 PaaS 平台的实现提供了一条全新的途径,也使为开发者提供更简洁的服务成为了可能。基于 Docker 容器,开发人员不再需要花费大量精力来处理各种开发、测试和生产环境之间的差异,而直接将应用环境迁移到 PaaS 平台的运行环境,不必担心各种依赖和配置问题。传统的 PaaS 是一个应用程序级别的隔离,在应用程序和应用程序之前是独立的,共享相同的运行时环境。与此同时,由于更加细化的资源粒度,使得同样的 PaaS 平台可以同时运行更多的应用程序。但 PaaS 平台也存在问题,由于平台与其运行环境息息相关,因此必须为每个平台定制代码,这对它的通用而言是一个极大的阻碍[10]。

(4)进程层面的隔离方式—容器。

因为应用程序和执行环境是相关的,所以应用程序和执行环境可以打包在一起,因此应用容器技术解决 PaaS 平台的应用程序必须依赖于其执行环境的问题。容器是一个应用程序,它的一系列操作环境依赖于封装在盒子里的一个“盒子”里,以共享相同的操作系统内核,使用操作系统内核的一些特性来实现资源隔离,容器技术是一个过程级隔离。容器技术不仅解决了环境依赖的问题,而且使隔离的粒度进一步细化、容器生成和销毁更快(第二级)。Docker 作为轻量级虚拟化技术的代表,是 LXC 技术的扩展,被认为是虚拟化领域的一次革新。

因此,纵观整个云计算技术开发过程,最重要的主线之一是越来越深入、更细致的资源粒度,管理越来越方便。如果云计算的未来是一种公共资源,那么它所驱动的业务无疑具有一个更细粒度和更可度量的计算单元特性的通用执行环境。

2 虚拟化容器技术 Docker

2.1 云计算的分层

云计算总体分为三层,即 IaaS、PaaS、SaaS。

IaaS:Infrastructure-as-a-service,基础设施即服务,主要为用户提供通过互联网获得完善的计算机基础设施服务。

PaaS:Platform-as-a-servers,平台即服务,主要为用户提供可以访问的完整或部分的应用程序开发。

SaaS:Software-as-a-service,软件即服务,主要为用户提供完整的可直接使用的应用程序。

2.2　Docker 容器技术的优势

Docker 是一种基于 lxcbased 的高级容器引擎开放源码的 PaaS 提供商 dotCloud。源代码托管在 Github 上,基于 go 语言,并遵循 Apache 2.0 协议开源。Docker 以 Linux 容器(LXC)技术为基础,主要功能是通过实现对 LXC 的进一步封装,使得对容器的操作变得更加简便,并且让用户不再需要关心容器的管理[11-13]。

与基于 Xen、KVM 虚拟机相比,其优势在于以下两点:

(1)启动速度快。虚拟机无法做到秒级启动是因为受限于操作系统的启动时间。但 Docker 完美地避开了这一问题。Docker 通过利用宿主机的系统内核,在几秒钟内创建大量容器,实现了秒级启动。由于虚拟机与 Docker 的启动速度是在数量级上的差距,实现更轻量级的虚拟化,方便快速部署。

(2)资源利用率高,性能开销小。虚拟化会比容器消耗更多的资源,Docker 除去占用的系统资源,剩余资源的 99% 都将提供给用户使用,极大地提高了资源利用率。

3　轻量级云存储系统

3.1　系统架构

最小的云存储系统架构由两个标准的 x86 服务器和一个千兆交换机组成。市场标准的 x86 服务器不仅满足了云存储磁盘的需要,而且还作为一个运行主机的 Docker 服务运行主机,以下简称为市场位 x86 服务器。系统架构如图 1 所示。

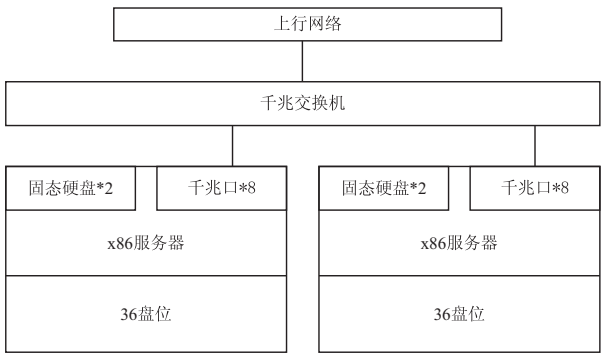

图 1　最小的云存储系统宏观架构

为了在发生单点故障后,系统仍能继续提供服务,该系统采用 2 台宿主机。当系统出现单点故障时,一定会出现数据丢失的情况,所以还需要对双宿主机最小系统的数据节点查找方式进行改进。单点故障发生时数据有冗余,系统保证另一台宿主机中数据的完整性,也同时为该节点继续提供完整的存储服务。

每台宿主机上,分别启动一个元数据服务器容器(meta data server,MDS)和一个数据节点(data node,DN)容器,形成由 2 个 MDS 容器和 2 个 DN 容器组成的云存储系统,2 台 MDS 之间的心跳通过交换机完成,简化了安装部署。

3.2　网络通信

容器部署采用桥接的网络模式(所谓桥接模式,即宿主机内部存在一个虚拟二层交换机,能够将容器中的虚拟网口和宿主机的网口在局域网中被平等地对待),使容器中的虚拟网口拥有了独立的 IP 地址,在网络中的行为等同于一台虚拟服务器[14]。桥接网络模式拓扑如图 2 所示。

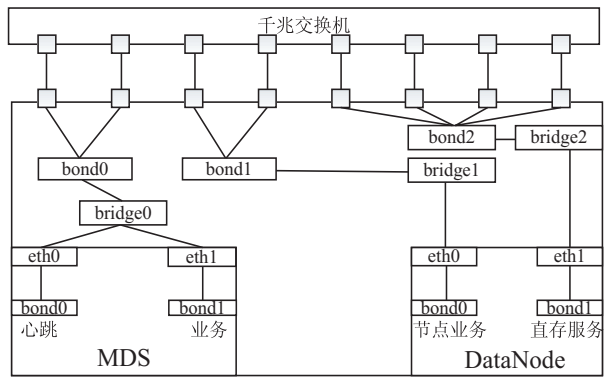

图 2　单节点结构与网络桥接方案

在宿主机系统启动时,其默认带有的 8 个千兆网口,会按照策略进行绑定的动作,可按照 2,2,4 的方式绑定成 3 个 bonding 口。这 3 个 bonding 口会分别挂在 3 个虚拟的桥接口下面,作为虚拟网桥的对外出口。

在主机中,MDS 和 DN 被部署在一个 Docker 镜像模式的容器中。当系统第一次启动时,MDS 和 DN 容器会自动开始等待用户配置初始化参数(如 IP 地址)。未配置的 MDS 通常有一个默认的出厂 IP。

MDS 容器在启动时,会创建 2 个虚拟口,这 2 个虚拟口都挂在宿主机第 1 个 bonding 口 bond0 下面,即 MDS 将 bond0 作为对外出口。MDS 容器中,MDS 可使用 bonding 口作为其心跳和业务网口,故将其虚拟的 eth0 和 eth1 两个网口分别绑定为 bond0 和 bond1 口,作为其心跳口和业务口。当然亦可直接指定 eth0 和 eth1。

宿主机中的 DN 容器在启动时,会创建 2 个虚拟口,这 2 个口分别挂在宿主机的第 2 个 bonding 口 bond1 和第 3 个 bonding 口 bond2 下面,即 DN 容器会用到 6 个物理口做成的 2 个 bonding 口作为对外通讯的接口。

DN 容器存在 2 个虚拟网口,分别是在外部绑定了 2 个网口的 eth0 和 4 个网口的 eth1,其中 eth1 给 DN 进程使用,用于 DN 的主要业务,即云存储的数据节点业务;eth0 可供其他服务使用,如监控中的数据流与指令流的传输等,用于提供云存储的直存服务

(CDS)。

3.3 系统性能

3.3.1 测试环境

· 2 台宿主机分别提供 1 个 MDS 容器。2 个 MDS 容器分为主备,并且同时处于工作状态。

· 系统的元数据信息保存于宿主机内部的 Intel128G SSD 中。

· SSD 挂载方式:开启“nobarrier”标识。

· 每台宿主机配备 24 块机械硬盘,配备千兆存储网口,通过视频模拟器同时输入 400 路 2 Mbps 视频,每个视频按 512 MB 文件写入。

存储节点采用 Erasure code 冗余备份机制,采用 1+1,4+1,8+2 节点间冗余容错策略,也就是说损失任意 1 或 2 个节点,客户端不会有感知,服务不会停止,数据不会丢失。按照 1+1,4+1,8+2 写入情况下主备宿主 CPU,磁盘利用率,写入带宽。

3.3.2 测试结果

三种机制磁盘性能比较如表 1 所示。

表 1 三种机制磁盘性能比较

备份机制	1+1	4+1	8+2
写入带宽/(MB/s)	45.92	15.62	10.32
磁盘利用率/%	38.25	20.35	10.55

在 1+1 机制下写入带宽和磁盘利用率均最高,基本满足了用户对磁盘写入的需求。磁盘读取速度通常高于写入速度,文中不再赘述。

1+1、4+1、8+2 三种机制写入情况下 MDS 中 mysqld 服务占用 CPU 百分比如图 3 所示。

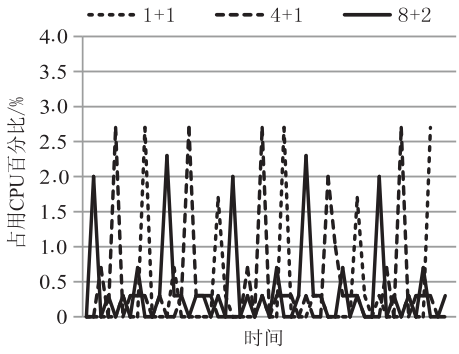

图 3 mysqld 占用 CPU 百分比

1+1、4+1、8+2 三种机制写入情况下 DN 占用 CPU 百分比如图 4 所示。

4 结束语

文中介绍了一种基于 Docker 容器技术的轻量级云存储系统架构,提出在系统中出现虚拟化环境下的网络通信改进策略。系统性能测试结果表明,该系统可以很好地满足一般用户对低成本、高效、数据可靠的云环境需求。

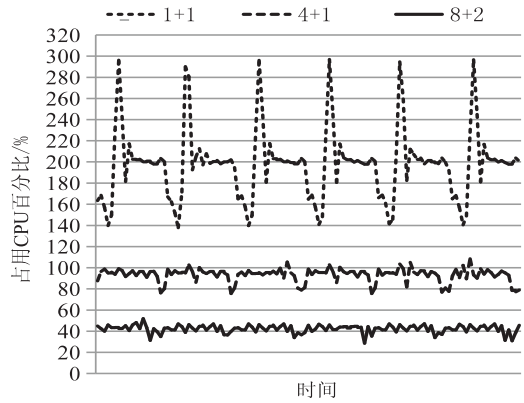

图 4 DN 占用 CPU 百分比

参考文献:

[1] 李 逦. 浅析云计算背景下云存储的优势与劣势[J]. 计算机光盘软件与应用,2013,16(23):18-19.

[2] 孔陶茹. 云存储应用的现状、挑战、展望、创新及探讨[J]. 物联网技术,2014,4(2):69-71.

[3] 肖 亮,李强达,刘金亮. 云存储安全技术研究进展综述[J]. 数据采集与处理,2016,31(3):464-472.

[4] 陈 全,邓倩妮. 云计算及其关键技术[J]. 计算机应用,2009,29(9):2562-2567.

[5] KHAN S,NAZIR B,KHAN I A,et al. Load balancing in grid computing:taxonomy,trends and opportunities[J]. Journal of Network & Computer Applications,2017,88:99-111.

[6] MELL P,GRANCE T. The NIST definition of cloud computing[J]. Communications of the ACM,2009,53(6):50.

[7] 崔 勇,宋 健,缪葱葱,等. 移动云计算研究进展与趋势[J]. 计算机学报,2017,40(2):273-295.

[8] 薛 涛,刘 龙. 云计算中虚拟机资源自动配置技术的研究[J]. 计算机应用研究,2016,33(3):759-764.

[9] 张笑燕,王敏讷,杜晓峰. 云计算虚拟机部署方案的研究[J]. 通信学报,2015,36(3):241-248.

[10] 俞乃博. 云计算 IaaS 服务模式探讨[J]. 电信科学,2011(S1):39-43.

[11] 汪 恺,张功萱,周秀敏. 基于容器虚拟化技术研究[J]. 计算机技术与发展,2015,25(8):138-141.

[12] 董 博,王 雪,索 菲,等. 基于 Docker 的虚拟化技术研究[J]. 辽宁大学学报:自然科学版,2016,43(4):327-330.

[13] ANDERSON C. Docker [software engineering][J]. IEEE Software,2015,32(3):102-c3.

[14] MIJUMBI R,SERRAT J,GORRICHO J L,et al. Network function virtualization:state-of-the-art and research challenges[J]. IEEE Communications Surveys & Tutorials,2017,18(1):236-262.