

DOI:10.19479/j.2095-719x.2403206

Ceph 在国产操作系统上的移植研究

许朝阳¹, 刘树东¹, 石春刚², 张艳¹

(1. 天津城建大学 计算机与信息工程学院, 天津 300384; 2. 中国航天科工三院 三〇四所, 北京 100074)

摘要:通过对中科方德与 CentOS 操作系统中的基础配置、宏定义和软件依赖包之间的异同分析,及基础环境要求、编译打包工具和软件源等移植基本需求的梳理工作,设计了移植环境及编译支持的配置方案,并根据 CentOS 操作系统中适配的 SRPM 文件,设计并编写了适配于中科方德国产操作系统的 SPEC 文件,完成了 Ceph 移植软件包的生成工作. 针对不同国产操作系统的特性,总结了国产操作系统中 Ceph 的移植要点. 部署测试及功能性测试结果表明,移植版 Ceph 功能完备,符合移植要求.

关键词:移植; 信息安全; Ceph; CentOS

中图分类号:TP311.54 **文献标志码:**A **文章编号:**2095-719X(2024)03-0206-07

Ceph Porting Study on Domestic Operating Systems

XU Zhaoyang¹, LIU Shudong¹, SHI Chungang², ZHANG Yan¹

(1. School of Computer and Information Engineering, TCU, Tianjin 300384, China; 2. The 304th Institute, The Third Academic of China Aerospace Science and Industry Corporation(CASIC), Beijing 100074, China)

Abstract: This paper mainly designs the configuration scheme of the porting environment and compilation support by analyzing the similarities and differences of the basic configuration, macro definitions and software dependency packages in the operating systems between ZhongKeFangDe and CentOS. It also sorts out the basic requirements of porting such as basic environment requirements, compilation and packaging tools as well as software sources. According to the SRPM file adapted to CentOS operating system, the SPEC file adapted to the domestic operating system of ZhongKeFangDe is designed and written, and the Ceph porting software package is completed. The main points of Ceph porting in domestic operating systems are summarized targeting at the characteristics of different domestic operating systems. The results of deployment test and functional test show that the transplanted Ceph is fully functional and meets the requirements of porting.

Key words: porting; information security; Ceph; CentOS

随着虚拟化、软件定义等 IT 技术的飞速发展,云计算、大数据和物联网产生的海量数据源源不断地涌向数据中心^[1],导致存储技术面临着严峻挑战. Ceph^[2-6]分布式存储技术的出现及发展则恰好解决了这一问题,其特有的高扩展性和高可用性^[7-8]使其一跃成为当前的主流分布式存储技术.同时,该技术可以提供文件存储、块存储和对象存储三大存储接口,满足了绝大多数企业的存储需求.随着大数据时代的到来,我国部分政企事业单位也纷纷要求部署 Ceph 分布式系统,以满足日益增长的数据存储需求.

然而当前时代“信息战”也正在悄无声息的展开,种种事件表明,信息安全应该是安全保障的制高点,是实现国家总体安全的重要基石^[9-10].在国家政企事业单位部署的 Ceph 分布式存储系统承担着国家重要信息的存储工作,保障其安全性是至关重要的.然而目前用于 Ceph 部署的安装包,仅支持国外主流发行版 Linux 操作系统,存在部分后门隐患,威胁国家信息安全.为解决上述问题,本研究主要在国产中科方德操作系统上进行 Ceph 的移植及部署测试工作,为重要信息的安全存储提供保障.

收稿日期:2022-10-18;修订日期:2023-03-30

基金项目:航天三院自主创新课题(20MK630620)

作者简介:许朝阳(1997—),男,河北石家庄人,天津城建大学硕士生.

通讯作者:刘树东(1965—),男,教授,博士,从事机器学习、软件工程研究. E-mail: liushudong@tcu.edu.cn

1 相关技术概述

1.1 Ceph 分布式存储技术

Ceph 分布式存储技术诞生于 2004 年, 其设计初衷是为了提供较好的性能、可靠性和可扩展性^[11-12]. 该项目起源于 Sage Weil 在攻读博士期间所做的工作, 在经过了数年的发展之后, 已经被应用在了各大云平台之上, 例如 Hadoop、Amazon S3 和 OpenStack 等^[13].

Ceph 是一个具有可靠性、可扩展性、统一性的分布式存储系统, 提供了对象(object)、块(block)及文件系统(file system)三种访问接口, 它们都通过底层的 LIBRADOS 库与后端的对象存储单元 (object storage device, OSD)交互, 实现数据的存储功能^[14-16]. 其整体架构主要由三个部分组成: 最核心的部分为 RADOS 对象存储系统^[17], 它是一个支持海量存储对象的分布式系统, 所有在 Ceph 系统中的数据最终都由这一层来存储; 其次为 LibRados 层, 该层使得本地或者远程应用可以通过网络来访问 RADOS 对象存储系统, 同时用户可以根据自己的需求调用 LibRados 来创建自己的应用; 最后为 Ceph 不同形式的存储接口, 其在 LibRados 的基础上为应用或客户端提供抽象层次更高、更便于使用的上层接口, 来满足绝大多数用户的需求. Ceph 架构如图 1 所示.

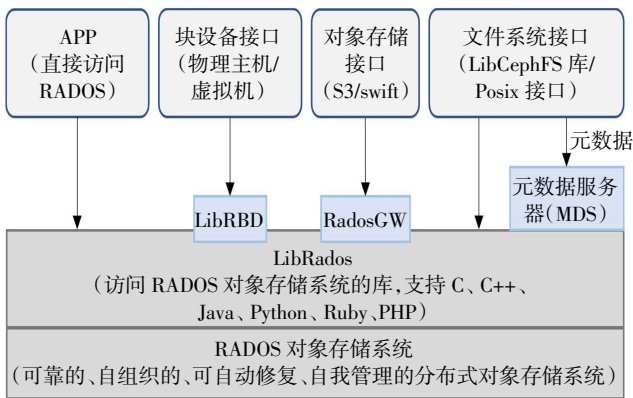


图 1 Ceph 架构

1.2 中科方德操作系统

中科方德是由中科院软件研究所研制, 重点应用于电子政务、国防军工、教育、能源交通等重点行业的操作系统. 产品广泛应用于全国数十个部委单位、各省市机关及企事业单位. 它以保证国家党政军及重大行业信息系统安全为己任, 努力打造值得用户信赖的服务器及桌面操作系统^[18].

中科方德操作系统可适配海光、兆芯、飞腾、龙芯、申威、鲲鹏等国产 CPU, 支持 x86、ARM、MIPS 等主流架构, 可良好支持台式机、笔记本、一体机及嵌入式设备等形态整机、主流硬件平台和常见外设, 具有良好的国产处理器适配性和通用性. 其操作方式、常用命令与主流发行版 Linux 系统基本一致, 安全性高, 可为信息安全保障工作提供强大助力.

2 Ceph 移植

在国产化移植过程中, 由于国家企事业单位的保密性要求, 许多服务器都被禁止接入外网, 因此只能采用 RPM 包安装的方式进行 Ceph 分布式存储系统的部署. 首先, 本研究通过对比 CentOS 与中科方德的异同, 配置中科方德系统上的移植环境; 然后, 根据 CentOS 的 SRPM 文件, 设计并编写适配于中科方德系统的 SPEC 文件, 完成 Ceph 源码在中科方德系统中的编译, 并将其打包为 RPM 安装包; 最后, 在中科方德系统上配置相应的软件安装依赖包, 完成整体的 Ceph 移植工作.

2.1 系统关键差异

本节研究了 CentOS7 与中科方德系统的差异, 并分析了这些差异对移植测试工作的影响, 例如与包管理有关的 YUM/DNF 工具、与软件编译有关的 GCC 及与部署有关的时间同步机制等. 在移植及测试中, 本研究针对这些差异调整了工具的使用方式、配置文件的内容及部署的操作. 经实验, CentOS 与中科方德操作系统的基本信息关键差异如表 1 所示.

表 1 CentOS 与中科方德系统基本信息的关键差异

配置名称	CentOS	中科方德
文件系统	XFS	ext4
时间同步	NTP/Chronyd	Chronyd
包管理工具	YUM	DNF/YUM
防火墙	Iptables	iptables/nftables
Git	1.8	2.18
GCC	4.8.5	8.3.1
Make	3.82	4.2

此外, 通过移植实验发现, 在中科方德系统中/usr/lib/rpm/macros 文件内的部分默认宏定义与 CentOS 中存在不同, 宏定义的主要区别之处如表 2 所示. 这些宏定义的不同会导致移植过程中部分链接的出错及 SPEC 文件的宏指令继承错误, 从而影响多个包的编译工作, 例如在执行 python 脚本文件时的“ambiguous python shebang”错误等. 为解决该问题, 本研究将中科

方德 macros 文件中的默认宏定义指向全部修改为 CentOS 中的配置。

表 2 宏定义区别

宏定义	CentOS	中科方德
__gpg	%{_bindir}/gpg2	/usr/bin/gpg2
__python	/usr/bin/python	/usr/bin/python3
__zstd	无	/usr/bin/zstd
__gem	无	/usr/bin/gem

值得注意的是,在中科方德系统中部分软件包的名称与 CentOS 系统存在差别,导致出现无法找到某个软件包及依赖条件判定错误的问题.例如在 Python 类中,部分 x-python 包在中科方德中为 x-python2 或者 x-python3.针对该问题,本研究梳理了两个系统的主要软件,筛选出不同名字的软件包,以此修改了 install 过程中不同软件包的名称及 SPEC 文件中软件包的依赖配置信息.除此之外,中科方德系统中还缺少部分系统与三方软件源中没有的软件包,为此,本研究从软件包网站获取相应的源码并在实验机中进行了包的移植.最后,本研究发现中科方德系统中部分软件包的版本过高(如 nspr、nss 等),与其余软件包不匹配从而导致编译错误.因此,本研究根据软件包的发布时间线将所有软件包进行了版本匹配并更新安装,以此来解决所有的软件包依赖问题,辅助完成 Ceph 整体的移植工作。

2.2 移植过程

本次移植选用的 Ceph 版本为 12.2.13(Luminous 长期支持版),国产操作系统为 X86_64 架构的中科方德 4.0 高可信服务器。

2.2.1 基本配置及编译支持

首先,针对中科方德系统下的移植工作进行环境的基本配置,在全新安装的中科方德服务器(硬盘剩余空间大于 50 G,运行内存大于 4 G)上,修改 ifcfg-ens33、resolv.conf、network 及 hostname 文件,以配置网络 ip、子网掩码、网关、DNS 及主机名,并使用 nmcli c reload 命令重启网络服务连接外网,完成网络配置工作.同时,使用 dnf install 命令进行 make 工具、rpmbuild 工具及 gcc 编译器的安装,以支撑 C/C++代码的编译、RPM 打包等工作。

其次,由于中科方德官方软件源无法支持所有软件依赖包的安装工作,因此本研究通过在/etc/yum.Repos.d 目录中添加配置文件 Nfs-Epel.repo 的方式进行 EPEL 三方软件源的配置,辅助解决了移植软件包依赖的问题。

最后,使用 dnf install 命令安装 zlib-devel、bzip2-

devel 等 Python 软件依赖包,从 Python 官网获取 python2.7 版本源码,并使用./configure --prefix=/usr/local/python2 及 make && make install 命令进行编译安装.修改~/.bash_profile 文件配置环境变量并配置软连接,完成 Python 环境的配置工作.至此,移植工作中的环境基本配置及编译支持工作完成。

2.2.2 SPEC 文件设计

在配置好中科方德系统的基本环境之后,下一步工作为制作 RPM 软件包.其中最关键之处在于 SPEC 文件的编写.SPEC 文件又称为软件包描述文件,该文件中包含了软件包的诸多信息,如软件包的名称、版本、类别、说明摘要、创建指令、安装操作及文件列表等.其信息涵盖了编译打包的整个过程,对于 Ceph 移植工作来说是至关重要的。

本研究所设计的 SPEC 文件总体包括了软件包基本信息、先决条件及输入和主要命令三大模块.其主要内容有版本(name,version,release)、其它软件有关的信息(group,license,URL,summary)、先决条件(prereq,buildrequires,requires,provides,obsoletes)、输入(source[#],patch[#])、BuildRoot 等.其包括了%prep、%patch、%build、%install、%clean、%files、%changelog、Scriptlets 几大阶段.其各个阶段属于顺序执行关系,即上一阶段的顺利成为为下一阶段的执行条件,本研究所设计的 SPEC 文件重要基本配置介绍如表 3 所示,模块逻辑关系如图 2 所示.对照 CentOS 版本的 SRPM 文件

表 3 SPEC 文件的重要基本配置介绍

模块	介绍
Source	源代码路径
ExclusiveArch	目标处理器架构
Patch	补丁包文件
PreReq	先决依赖
Requires	安装依赖
BuildRequires	编译依赖

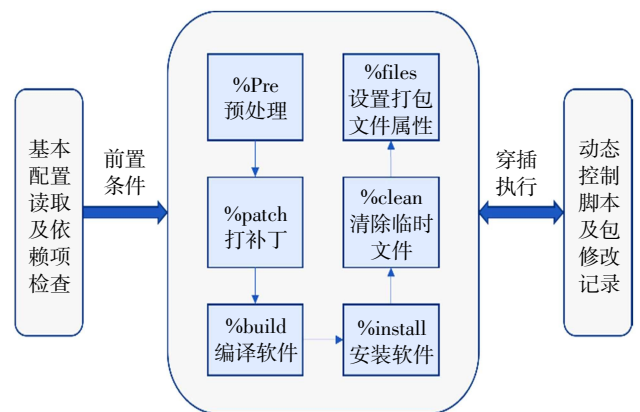


图 2 模块逻辑关系

中提取出的 SPEC 文件和系统间的重要差异,例如系统架构、依赖包名称等,需将脚本命令及配置信息写入到本文的 SPEC 文件中,来控制编译打包的整体流程。

本研究在软件包基本信息模块中,定义了软件名称、软件版本及软件摘要等基本说明信息,其关键配置内容如表 4 所示。

表 4 软件包基本信息关键配置

字段	内容
Name	ceph
Version	12.2.13
Release	0%{?dist}
Summary	NFS Ceph file system
Group	System/Filesystems

在先决条件及输入模块中,定义了支持架构、软件依赖包、源码输入、补丁包等信息。其中将 BuildRequires 及 Requires 字段内容修改为了中科方德操作系统中的软件依赖包名称,其关键配置内容及部分软件依赖包名称如表 5 所示。

表 5 先决条件及输入模块关键配置内容

字段	内容
ExclusiveArch	x86_64 aarch64 ppc64 ppc64le
BuildRequires	python3-prettytable、python2-devel 等
Source0	/opt/rpmbuild/SOURCES/ceph-12.2.13.tar.gz
Requires	python3-pecan、python2-jinja2 等

在主要命令模块中,共分为%prep、%build、%install、%clean、%files 几大阶段。其中:% prep 阶段负责编译前的源码解压工作;%build 阶段负责编译工作,其中包括编译引用、编译参数设置等;%install 阶段负责将软件安装到虚拟的根目录中,其主要是为%files 阶段所服务的;%clean 阶段主要负责编译安装后的临时文件清理工作;%files 阶段定义了打包文件的列表、目录、权限等属性。本研究中的 SPEC 文件主要命令模块各个部分的关键脚本如表 6 所示。

表 6 主要命令模块关键脚本

阶段	关键脚本	功能
%prep	autosetup -p1 -n ceph-12.2.13	解压源码代码
%build	cmake ..\-DWITH_BOOST_CONTEXT=ON	针对目标机的 x86_64 架构,添加 Boost.Context 库
%install	make DESTDIR=%{buildroot} install	为避免%files 阶段寻找文件出错,安装路径设置为统一路径
%clean	rm -rf %{buildroot}	清理产生的临时文件
%files	%{*_bindir}/crushtool 等	设置文件生成目录、文件对应权限等属性

2.2.3 编译及打包

首先,获取 RedHat 系列的 Ceph Luminous 源码,

并创建打包目录 rpmbuild,其各个子目录的功能如表 7 所示。

表 7 rpmbuild 目录介绍

子文件夹	名称	作用
SPECS	spec 文件目录	存放配置(.spec)文件
SOURCES	源代码目录	存放源码包和 patch 补丁文件
BUILD	编译目录	源代码解压至此并完成编译
BUILDROOT	安装目录	保存%install 阶段安装的文件
RPMS	RPM 包目录	生成/存放二进制 RPM 安装包
SRPMS	SRPM 包目录	生成/存放 SRPM 文件

其次,将 2.2.2 中设计好的 SPEC 文件拷贝到 SPECS 子目录中,使用 rpmbuild -bp ceph.spec 命令进行预处理操作,在提示依赖包错误之后,根据提示使用 dnf install xx 命令对 Cython、python2-devel 等软件包进行安装以通过预处理阶段的依赖检查。

在预处理阶段处理成功之后,使用 rpmbuild -ba ceph.spec 命令开始进行编译打包工作,等待命令执行完毕之后,便可在~/rpmbuild/RPMS 目录下生成一系列中科方德版本的 Ceph 安装包。中科方德操作系统下的 Ceph 安装包列表如图 3 所示。

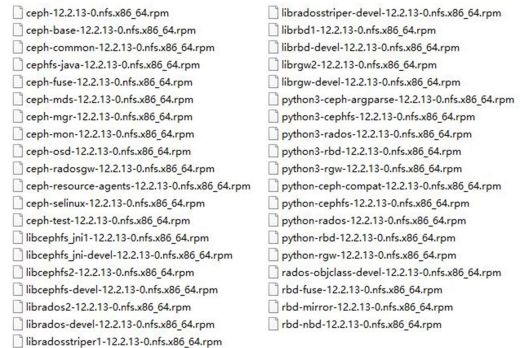


图 3 中科方德操作系统下的 Ceph 安装包

最后,在目标机安装 Ceph 之前,需准备一系列的安装依赖包。本研究在一个全新安装的中科方德服务器中,利用 rpm -q * 命令的提示,使用 dnf install --downloadonly --downloadaddir=/tmp/** 命令逐一进行安装依赖包的获取,最终获取的 Ceph 安装依赖包列表如图 4 所示。

2.3 Ceph 在国产操作系统中的移植要点

通过 Ceph 在中科方德操作系统上的移植研究,总结了国产操作系统中 Ceph 的移植要点,主要包括以下几个方面。

(1)根据 CPU 架构信息、系统内核版本等信息判断该国产操作系统属于何种系列以获取相应的 Ceph 软件源码。并根据系列信息对系统进行相应的系统配

```

gperftools-libs-2.7-9.el8.x86_64.rpm
hamcrest-core-1.3-24.nfs.noarch.rpm
ip-1.5-12.nfs.x86_64.rpm
junit-4.12-9.nfs.noarch.rpm
leveldb-1.22-1.el8.x86_64.rpm
libunwind-1.3.1-3.el8.x86_64.rpm
htng-ust-2.8.1-11.nfs.x86_64.rpm
oniguruma-6.8.2-1.nfs.x86_64.rpm
python2-babel-2.5.1-9.module.el8.1.0+219+cf9e6ac9.noarch.rpm
python2-jinja2-2.10.10.nfs4.noarch.rpm
python2-markupsafe-0.23-20.nfs.x86_64.rpm
python2-prettytable-0.7.2-3.23.noarch.rpm
python2-pytz-2017.2-12.nfs.1.noarch.rpm
python2-setuputils-39.2.0-7.nfs.noarch.rpm
python3-babel-2.5.1-7.nfs4.noarch.rpm
python3-beautifulsoup4-4.6.3-2.el8.1.noarch.rpm
python3-cheroot-8.5.2-1.el8.noarch.rpm
python3-cherryypy-18.4.0-1.el8.noarch.rpm
python3-click-6.7-9.nfs.noarch.rpm
python3-flask-0.12.2-3.nfs.noarch.rpm
python3-itsdangerous-0.24-14.nfs.noarch.rpm
python3-jaraco-6.2.el8.noarch.rpm
python3-jaraco-functools-2.0-4.el8.noarch.rpm
python3-jinja2-2.10-10.nfs4.noarch.rpm
python3-logutils-0.3.5-11.el8.noarch.rpm
python3-lxml-4.2.3-2.nfs.x86_64.rpm
python3-mako-1.0.6-13.nfs.noarch.rpm
python3-markupsafe-0.23-20.nfs.x86_64.rpm
python3-more-iterutils-7.2.0-3.el8.noarch.rpm
python3-pecan-1.3.2-9.el8.noarch.rpm
python3-portend-2.6-1.el8.noarch.rpm
python3-prettytable-0.7.2-14.nfs.noarch.rpm
python3-simplegeneric-0.8.1-17.el8.noarch.rpm
python3-singledispatch-3.4.0.3-18.el8.noarch.rpm
python3-tempora-1.14.1-5.el8.noarch.rpm
python3-trustme-0.6.0-4.el8.noarch.rpm
python3-waitress-1.2.1-2.el8.1.noarch.rpm
python3-webob-1.8.5-1.el8.1.noarch.rpm
python3-webtest-2.0.33-1.el8.noarch.rpm
python3-werkzeug-0.12.2-4.nfs.noarch.rpm
python3-zc-lockfile-2.0-2.el8.noarch.rpm
resource-agents-4.1.1-33.nfs.x86_64.rpm
socat-1.7.3.2-6.nfs.x86_64.rpm
xmllint-1.6.1-11.el8.x86_64.rpm

```

图4 Ceph安装所需依赖包

置,例如网络配置、宏定义配置、软件源配置等。

(2)针对 RedHat 系列国产操作系统,采用rpmbuild 工具对 Ceph 源码进行编译并将其打包为 rpm 格式的安装包,最终使用 rpm 命令将其安装到目标机器中。针对 Debain 及 Ubuntu 系列国产操作系统,则采用 build 和 dpkg 等工具将其生成为 deb 格式的安装包,并采用 dpkg 命令将其安装到目标机器中。

(3)在软件依赖包方面,首先运行 Ceph 源文件中的 install-deps.sh 文件来自动安装所需要的软件依赖包,如仍提示缺少依赖,则根据提示继续使用 dnf/yum、apt-get 或者源码编译的方式进行手动安装。除此之外,不同操作系统之间的基础软件包数量、软件依赖包名称及版本会有所不同,应根据名称对应关系及版本对应关系进行相应配置。

(4)在 RedHat 系列国产操作系统的 SPEC 文件设计中,需针对不同 CPU 架构在 ExclusiveArch 字段添加相应信息,并根据不同系统名称及版本修改 if 判断。

3 移植效果测试

在 Ceph 安装包及安装依赖包全部准备完成之后,将其安装到中科方德内网服务器上,对其进行手动部署并连接 NFS 服务器进行完备性测试。

3.1 基础组件部署及测试

本测试在将所有安装包安装完毕之后,全程采用手动的方式进行部署。首先编写 ceph.conf 配置文件,设置密钥环,部署 MON、MGR、OSD 三大基础模块,其次分别部署文件存储接口(CephFS)、块设备接口(RBD)、对象存储接口(RGW)三大接口,并测试其功能及完备性。三大基础模块部署如图5所示,从图5可以看出,集群 health 状态显示“HEALTH_OK”,services 模块显示正常,MON、MGR 及 OSD 基础模块部署成功。

```

[root@localhost ceph]# ceph -s
cluster:
  id:          ef3bc1a1-03dd-4f14-b4cd-91af305c70f4
  health:     HEALTH_OK

services:
  mon: 1 daemons, quorum Ceph
  mgr: Ceph(active)
  osd: 1 osds: 1 up, 1 in

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0B
  usage:   1.00GiB used, 7.00GiB / 8.00GiB avail
  pgs:

```

图5 三大基础模块

(1)文件存储接口。在部署 CephFS 之前,需要首先启动 MDS(metadata-server 元数据服务器)的守护进程用来管理元数据,为 CephFS 的运行提供保障,部署图如图6所示。从图6可以看出,mds 显示为“active”,运行正常。

```

[root@localhost x86_64]# ceph -s
cluster:
  id:          1b43044c-7601-4e1e-add1-c1574e380faa
  health:     HEALTH_OK

services:
  mon: 1 daemons, quorum Ceph
  mgr: Ceph(active)
  mds: cephfs-1/1/1 up {0=Ceph:up:active}
  osd: 1 osds: 1 up, 1 in

data:
  pools:   2 pools, 32 pgs
  objects: 21 objects, 2.19KiB
  usage:   1.00GiB used, 6.99GiB / 8.00GiB avail
  pgs:     32 active+clean

```

图6 启动 MDS 服务

MDS 成功启动之后,创建数据与元数据资源池并对 CephFS 进行挂载。测试结果表明,目录挂载成功,文件存储接口运行正常,效果图如图7所示。

```

tmpfs on /var/lib/ceph/osd/ceph-0 type tmpfs (rw,relatime)
ceph-fuse on /mnt/mycephfs type fuse.ceph-fuse (rw,nosuid,nodev,relatime,
user_id=0,group_id=0,allow_other)
[root@localhost x86_64]# cp ~/anaconda-ks.cfg /mnt/mycephfs/
[root@localhost x86_64]# ls /mnt/mycephfs/
anaconda-ks.cfg

```

图7 CephFS 挂载成功

(2)块设备接口。在三大基础模块的基础上,无需启动其余服务,便可启用 RBD 接口。本测试为 RBD 创建资源池 pool,在 pool 的基础上创建镜像文件,再使用内核模块对镜像文件进行挂载。镜像文件挂载成功之后,对块设备进行分区格式化,最后新建数据目录并将其挂载到分区上。测试结果表明,目录挂载成功,

块设备接口运行正常,效果图如图 8 所示.

```
tmpfs on /var/lib/ceph/osd/ceph-0 type tmpfs (rw,relatime)
/dev/rbd0p1 on /mnt/rbdtest type xfs (rw,relatime,attr2,inode64,sunit=8192,
swidth=8192,noquota)
[root@localhost x86_64]# cd /mnt/rbdtest && cp ~/anaconda-ks.cfg /mnt/rbdte
st/
[root@localhost rbdtest]# ls
anaconda-ks.cfg
```

图 8 RBD 分区挂载成功

(3)对象存储接口. 在三大基础模块的基础上,手动部署对象存储接口 RGW, RGW 服务启动之后的 Ceph 集群状态如图 9 所示. 由图 9 可以看出, rgw 显示为“active”, 服务启动成功.

```
[root@localhost ceph]# ceph -s
cluster:
  id:          ef3bclal-03dd-4f14-b4cd-91af305c70f4
  health:      HEALTH_OK

services:
  mon: 1 daemons, quorum Ceph
  mgr: Ceph(active)
  osd: 1 osds: 1 up, 1 in
  rgw: 1 daemon active

data:
  pools:   4 pools, 32 pgs
  objects: 171 objects, 1.09KiB
  usage:   1.00GiB used, 6.99GiB / 8.00GiB avail
  pgs:     32 active+clean

io:
  client:  16.2KiB/s rd, 0B/s wr, 24op/s rd, 16op/s wr
```

图 9 启动 rgw 对象存储服务

在 RGW 服务启动之后,使其对接 S3(AWS 的标准云存储服务)测试对象存储接口移植的基础功能的完备性. 在 RGW 的基础上创建 S3 用户,配置CivteWeb 服务器,创建 Bucket,即可使用 s3cmd 命令对对象存储接口进行读写操作. 存取数据测试结果表明, 对象存储接口运行正常,效果图如图 10 所示.

```
[root@Ceph ceph]# s3cmd ls s3
[root@Ceph ceph]# s3cmd mb s3://demo 创建bucket
Bucket 's3://demo/' created
[root@Ceph ceph]# cd ~ 通过s3向bucket中存数据
[root@Ceph ~]# s3cmd put anaconda-ks.cfg s3://demo
upload: 'anaconda-ks.cfg' -> 's3://demo/anaconda-ks.cfg' [1 of 1]
1848 of 1848 100% in 1s 1013.44 B/s done 向外取数据
[root@Ceph ~]# cd /mnt && s3cmd get s3://demo/anaconda-ks.cfg
download: 's3://demo/anaconda-ks.cfg' -> './anaconda-ks.cfg' [1 of 1]
1848 of 1848 100% in 0s 460.03 kB/s done
[root@Ceph mnt]# ls
anaconda-ks.cfg hgfs s3cmd
```

图 10 通过 s3cmd 进行读写操作

3.2 功能性测试

3.2.1 NFS 连接测试

网络文件系统(network file system, NFS)^[19]可以通

过网络让不同的机器、不同的操作系统之间共享彼此的文件. Ceph 中的 RGW 及 CephFS 为其提供了相应的接口, 可以使用户通过 NFS 客户端来访问 Ceph 中的数据, 提高企业工作效率. 本节通过将 RGW 及 CephFS 导出为 NFS 文件接口, 进一步验证移植后的 Ceph 分布式存储系统有关 RGW 及 CephFS 模块的完备性.

目前将 Ceph 导出为 NFS 有两种方式: 一种为 RGW 对象存储接口导出方式, 其通过调用 librgw2 将 NFS 协议转义为 S3 协议, 再通过 RGW 将数据存入到 Ceph 中; 另一种为 CephFS 接口导出方式, 其通过调用 libcephfs2 将 NFS 协议转义为 CephFS 协议从而将数据存入到 Ceph 中. 部署 NFS 后对两种导出方式进行测试, 挂载目录如图 11 和图 12 所示. 可以看出, 目录挂载成功, 运转正常, 功能完备.

```
[root@Ceph rgw-nfs]# tree
├── rgw
│   └── demo
│       └── anaconda-ks.cfg
└── 2 directories, 1 file
```

图 11 NFS 客户端成功挂载 RGW 接口

```
[root@Ceph cephfs-nfs]# tree
├── mycephfs
│   └── anaconda-ks.cfg
└── 1 directory, 1 file
```

图 12 NFS 客户端成功挂载 CephFS 接口

3.2.2 底层存储可靠性测试

底层存储可靠性主要通过节点数据是否高可用来进行判断, 而节点数据是否高可用则通过硬盘故障实验和节点故障实验来测试. Ceph 集群通常设置为两副本或者三副本模式, 将 RBD 挂载到三个节点上, 通过断电和拔出硬盘操作来模拟单盘故障和节点故障测试文件是否可以继续正常读写. 节点数据高可用测试结果如表 8 所示, 测试结果表明, 移植后的 Ceph 集群底层存储可靠性良好.

表8 节点数据高可用测试

故障	操作	验证	结果
单盘故障	拔出硬盘	对比操作前后挂载目录下文件是否可读写	可读写且未损坏
节点故障	断电	对比操作前后挂载目录下文件是否可读写	可读写且未损坏

4 结 语

本文针对 Ceph 分布式存储技术在国产操作系统上的移植进行了研究,并在其内网服务器上通过手动部署的方式对移植包进行了完备性及功能性测试.测试结果表明,移植后的 Ceph 分布式存储系统运行正常,功能完备,符合移植要求,可为国家信息安全保障及存储性能提升工作提供强大助力.

参考文献:

- [1] 朱盼盼,张 彤,郑宇宇,等. 分布式存储系统中纠删码数据修复算法优化与实现[J]. 计算机应用研究, 2020,37(S1): 140-142.
- [2] WEIL S A, BRANDT S A, MILLER E L, et al. Ceph: a scalable, high-performance distributed file system[C]//Proceedings of the 7th Symposium on Operating Systems Design and Implementation. Berkeley: USENIX, 2006: 307-320.
- [3] 柯文龙,王 勇,叶 苗,等. Ceph 云存储网络中一种业务优先级区分的多播流调度方法[J]. 通信学报, 2020,41(11): 40-51.
- [4] 李新鹏,王 勇,叶 苗. Ceph 云存储中基于强化学习的 QoS 优化[J]. 计算机工程与设计, 2021,42(2): 455-461.
- [5] CHUM S, PARK H, CHOI J. Supporting SLA via adaptive mapping and heterogeneous storage devices in Ceph[J]. Electronics, 2021, 10(7): 847-860.
- [6] 张泽军. 基于 Ceph 块存储的高可用 ISCSI 研究与应用[D]. 成都: 电子科技大学, 2019.
- [7] 张 晓,张思蒙,石 佳,等. Ceph 分布式存储系统性能优化技术研究综述[J]. 计算机科学, 2021,48(2): 1-12.
- [8] 黄遵祥,朱磊基,熊 勇. 一种基于双控节点的 Ceph 写性能优化方法[J]. 中国科学院大学学报, 2022,39(6): 817-826.
- [9] 雷 蕾,蔡权伟,荆继武,等. 支持策略隐藏的加密云存储访问控制机制[J]. 软件学报, 2016,27(6): 1432-1450.
- [10] 任晓刚. 数字政府建设进程中的安全风险及其治理策略[J]. 求索, 2022(1): 165-171.
- [11] CHEN Y, MAO Y C. Automatic tuning of Ceph parameters based on random forest and genetic algorithm[J]. Journal of Computer Applications, 2020,40(2): 347-351.
- [12] ZHANG X, WANG Y Q, WANG Q, et al. A new approach to double I/O performance for Ceph distributed file system in cloud computing[C]//2019 2nd International Conference on Data Intelligence and Security (ICDIS). Piscataway: IEEE, 2019: 68-75.
- [13] AGHAYEV A, WEIL S, KUCHNIK M, et al. File systems unfit as distributed storage backends: lessons from 10 years of Ceph evolution[C]//Proceedings of the 27th ACM Symposium on Operating Systems Principles. New York: Association for Computing Machinery, 2019: 353-369.
- [14] 陈 禹,毛莺池. 基于随机森林和遗传算法的 Ceph 参数自动调优[J]. 计算机应用, 2020,40(2): 347-351.
- [15] HUANG M, LUO L, LI Y, et al. Research on data migration optimization of ceph[C]//2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). Piscataway: IEEE, 2017: 83-88.
- [16] 王 勇,叶 苗,何 倩,等. 基于软件定义网络和多属性决策的 Ceph 存储系统节点选择方法[J]. 计算机学报, 2019, 42(2): 93-108.
- [17] 常 涛. Ceph 源码分析[M]. 北京: 机械工业出版社, 2016.
- [18] 中科方德. 关于方德[EB/OL]. (2022-06-06)[2022-06-14]. <http://www.nfschina.com/index.php?catid=5>.
- [19] 钟 林. 基于 NFS 的云存储网关关键技术研究及系统实现[D]. 广州: 华南理工大学, 2015.