

**2015** 级

《物联网数据存储与管理》课程

**实 验 报 告**

**姓 名 徐臻林**

**学 号 U201514868**

**班 号 物联网工程1501班**

**日 期 2018.04.15**

**目 录**

[一、实验目的 1](#_Toc478027105)

[二、实验背景 1](#_Toc478027106)

[三、实验环境 1](#_Toc478027107)

[四、实验内容 2](#_Toc478027108)

[4.1 对象存储技术实验 2](#_Toc478027109)

[4.2 对象存储应用实验 2](#_Toc478027110)

[五、实验过程 2](#_Toc478027111)

[六、实验总结 9](#_Toc478027112)

[参考文献 10](#_Toc478027113)

# 一、实验目的

1. 熟悉对象存储技术，代表性系统及其特性；

2. 实践对象存储系统，部署实验环境，进行初步测试；

3. 基于对象存储系统，架设实际应用，示范主要功能。

# 二、实验背景

对象存储是面向对象/文件的、海量的互联网存储，它也可以直接被称为“云存储”。

对象尽管是文件，它是已被封装的文件（编程中的对象就有封装性的特点），也就是说，在对象存储系统里，你不能直接打开/修改文件，但可以像ftp一样上传文件，下载文件等。

另外对象存储没有像文件系统那样有一个很多层级的文件结构，而是只有一个“桶”的概念（也就是存储空间），“桶”里面全部都是对象，是一种非常扁平化的存储方式。

其最大的特点就是它的对象名称就是一个域名地址，一旦对象被设置为“公开”，所有网民都可以访问到它；它的拥有者还可以通过REST API的方式访问其中的对象。因此，对象存储最主流的使用场景，就是存储网站、移动app等互联网/移动互联网应用的静态内容（视频、图片、文件、软件安装包等等）。

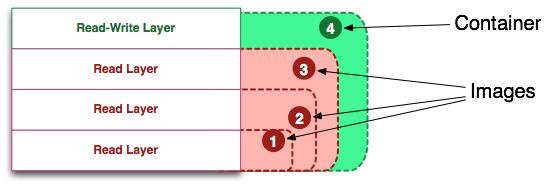
本次实验主要需要搭建一个网盘以及个人云，同时能够支持Thumbor的图片存储功能。

# 三、实验环境

实验的基础环境：

Linux系统与Docker容器

Windows系统与VirtualBox虚拟机

[](http://dockerone.com/uploads/article/20151103/d6ad9c257d160164480b25b278f4a2ad.png)

# 四、实验内容

## 4.1 对象存储技术实验

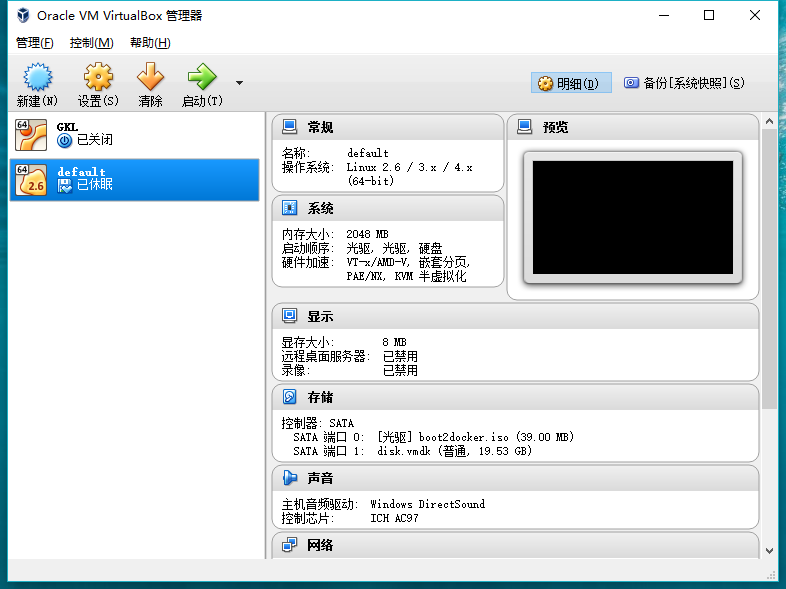
1. 在本机上搭建一个Minio服务器端，使用本机作为一个服务器，实现一个本地云。本机总存储空间就是服务器总空间，剩余空间即为服务器剩余空间。并且能够对服务器的用户名和密码进行更改，同时能够上传文件到云端，别的用户也能够从这个网站来把云端的文件下载下来。
2. 在本机开启一个Minio客户端。
3. 在Minio服务端开启的情况下，运行COSBench并跑测试程序来测试服务器性能。
4. 在本机搭建一个匿名的Mock-S3服务器，并在服务器开启情况下，运行COSBench来测试性能

## 4.2 对象存储性能分析

1. 在开启Minio服务器并测试COSBench的时候，观察在不同workers和不同sizes下服务器的读写能否成功及相关数据。如果有失败，对测试代码稍加更改，通过测试。
2. 在开启Mock-S3服务器并测试COSBench的时候，观察在不同workers和不同sizes下服务器的读写能否成功及相关数据。

# 五、实验过程

1.使用VirtualBox虚拟机创建Linux操作环境



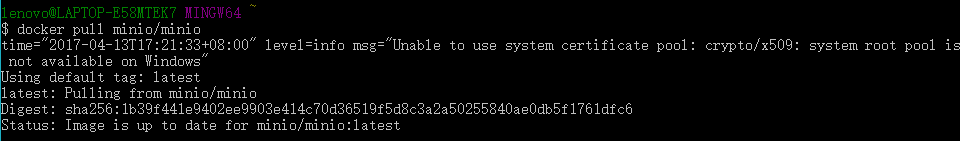
2.利用Docker pull镜像

相关操作为

docker pull minio/minio

docker pull asl/thumbor

完成后可查询到的结果如下图所示：



3.创建容器运行Minio

初次运行时需要利用下列指令进行创建

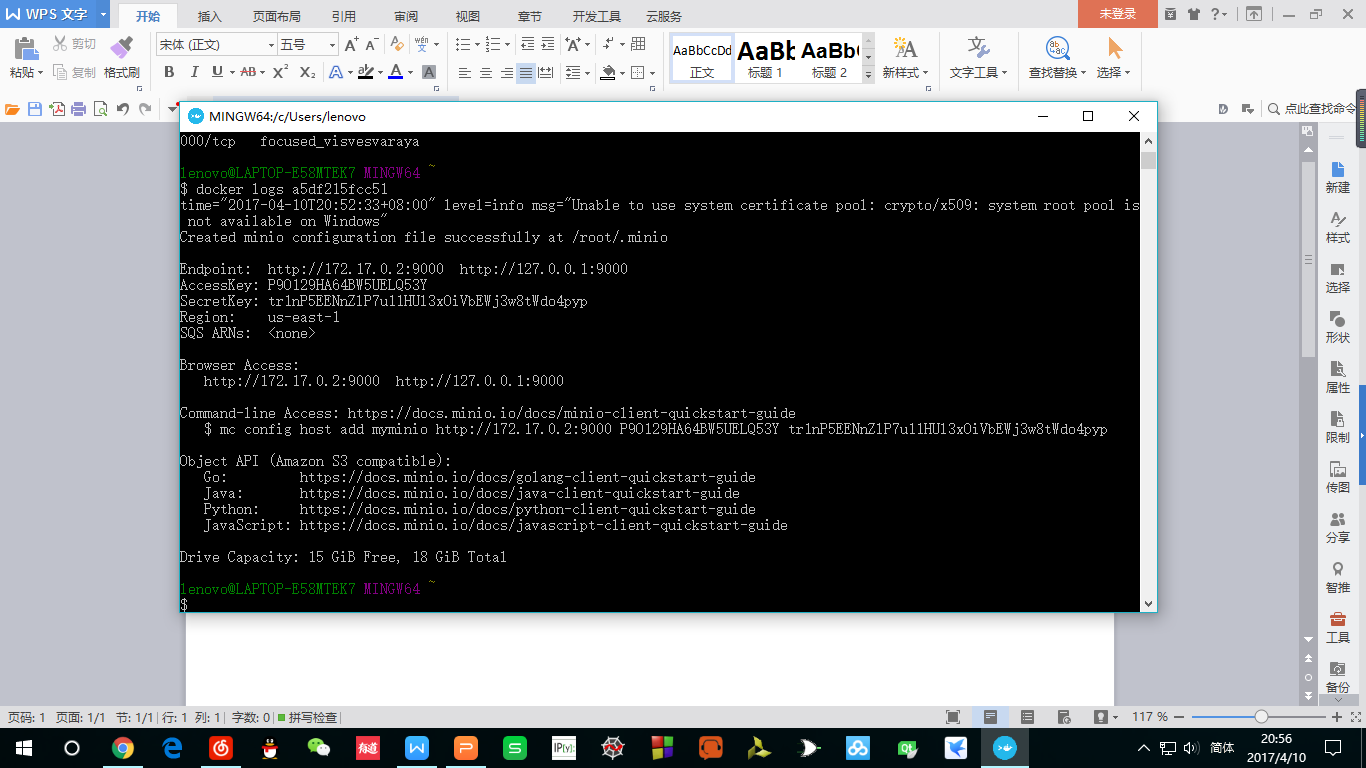
docker pull minio/minio

docker run -p 9000:9000 minio/minio server /export

由于之前已经创建过了Minio容器，再次登录时只用利用下面指令获取容器的ID即可运行

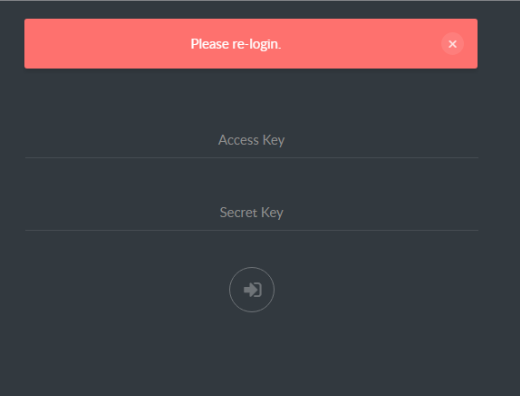
docker ps

docker logs

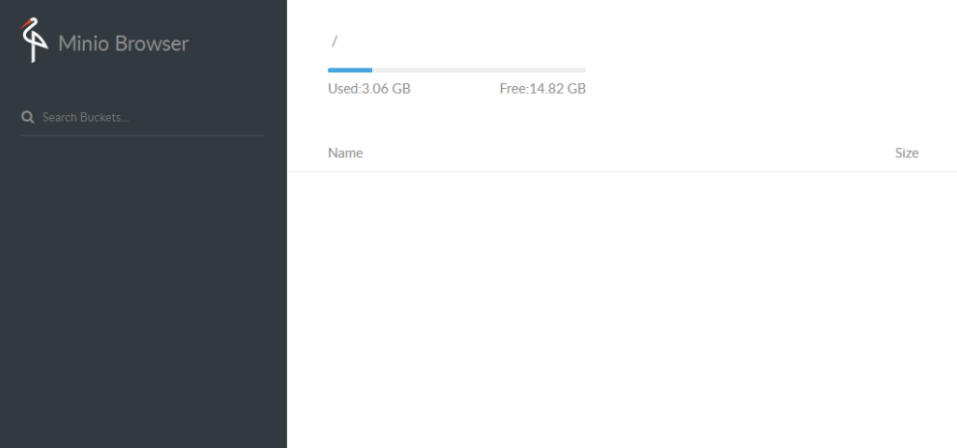


log后即可以得到相关的AccessKey,SecretKey,Region等信息

1. 访问172.17.0.2:9000，如下图所示：

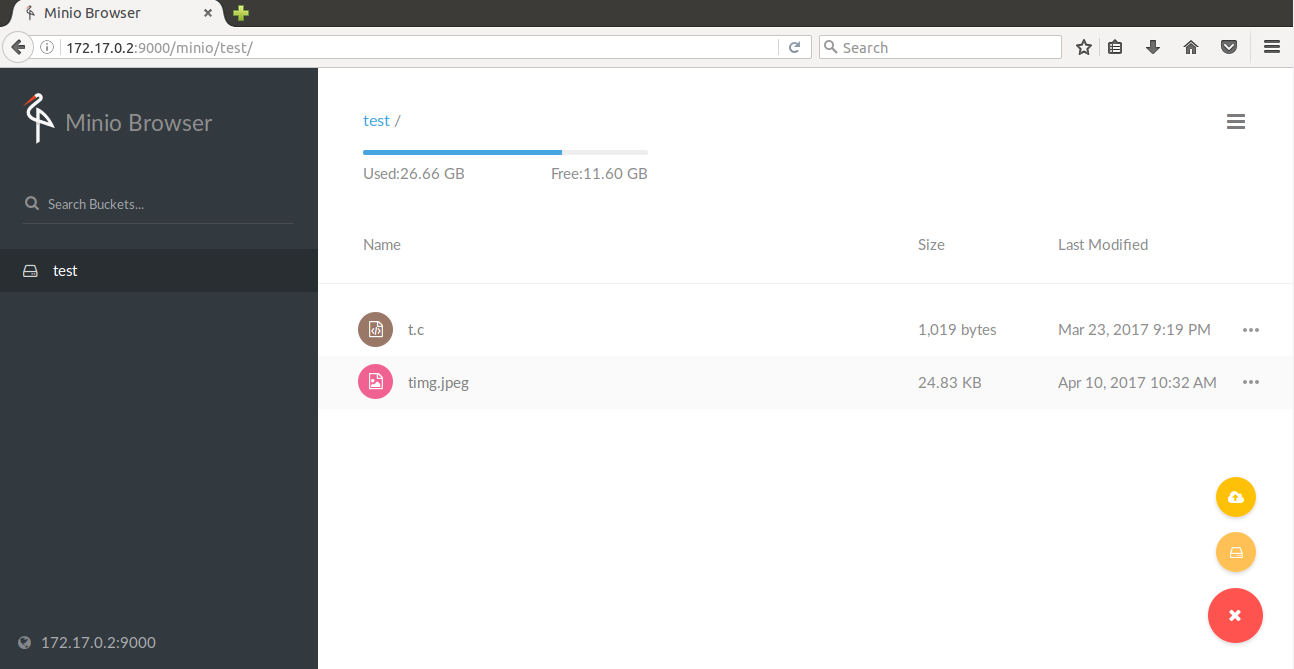


填入相关的AccessKey,SecretKey后即可进入Minio Browser,如下图所示：



5.创建Bucket并进行Upload 测试

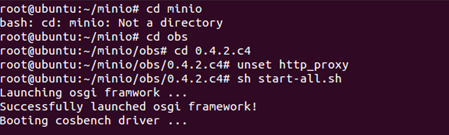
在Minio Browser中的操作结果如下图所示：



6.安装COSBENCH

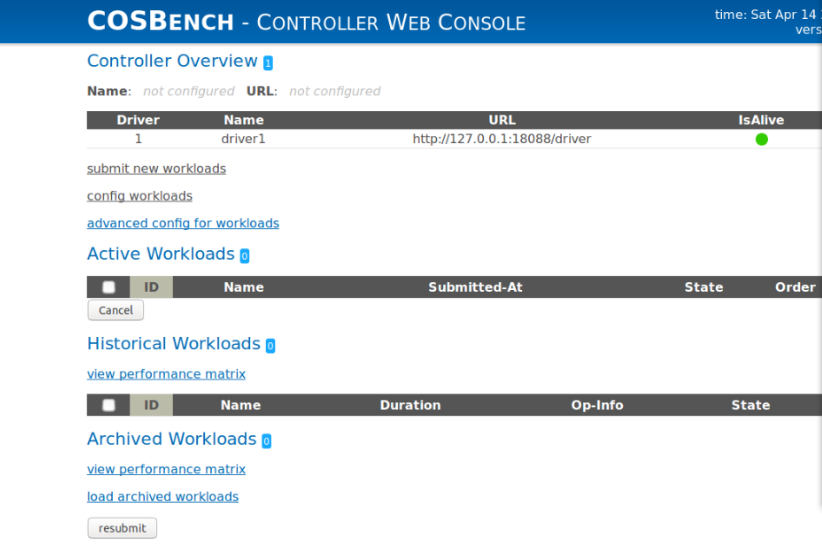
版本0.4.2.c4

终端进入COSBench所在文件夹，输入如下命令：

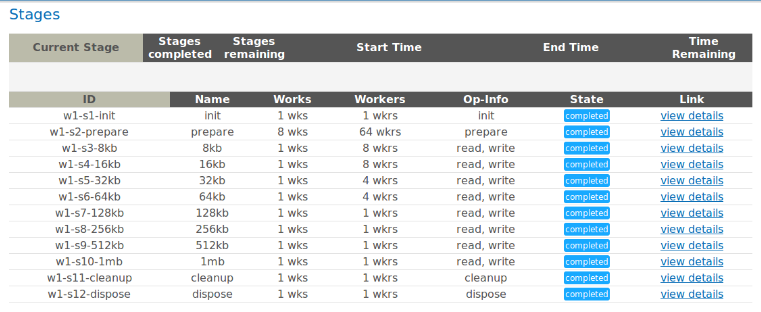


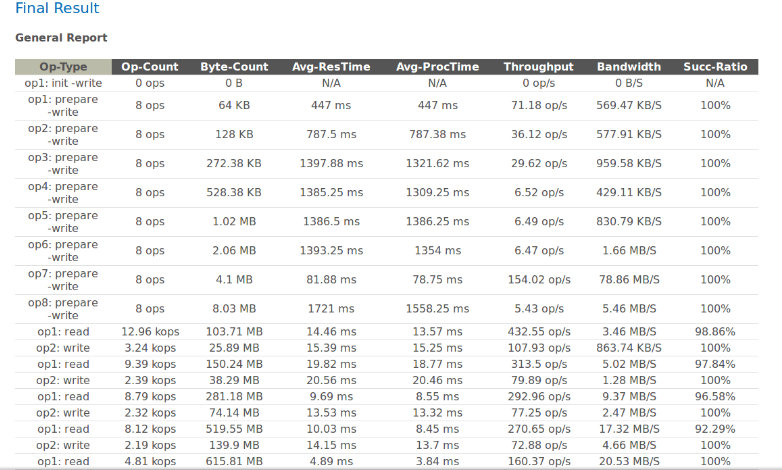
输入命令之后，进入网址<http://127.0.0.1:19088/controller/>

进入controller。

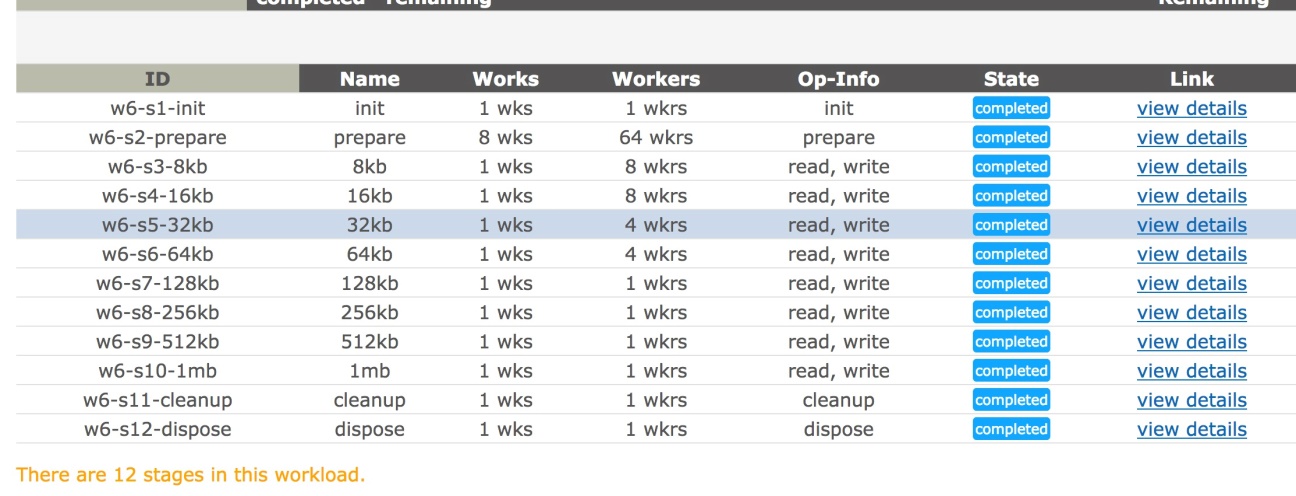


点击submit new workloads向对象存储中添加新的负载来进行接下来的测试。





7.配置Mock-s3

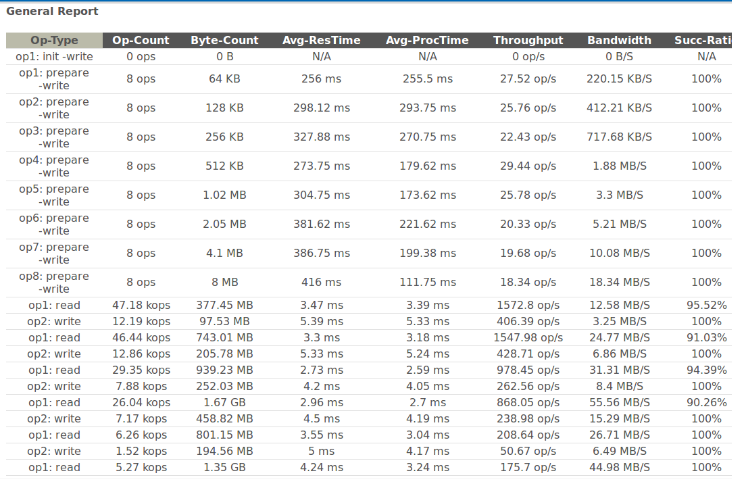


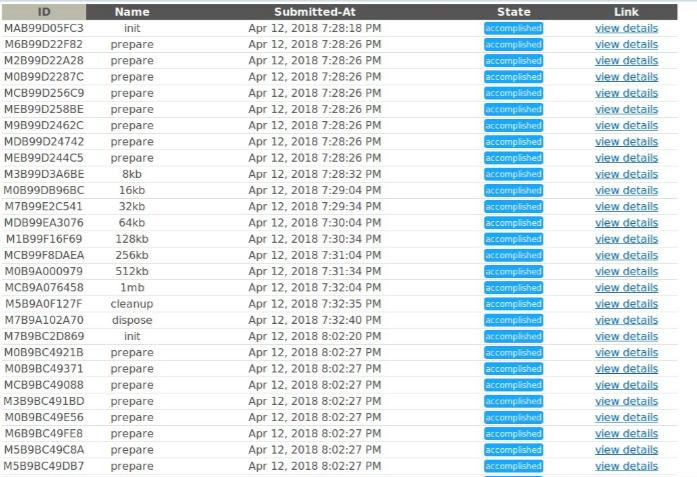
我们可以发现读的成功率并没有到100%，但是写的成功率已经是100%了

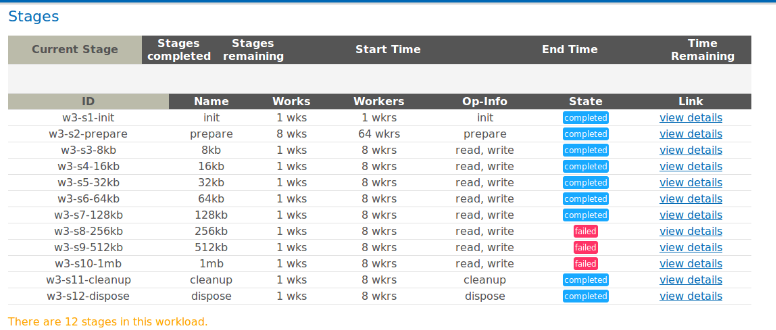
一开始的任务是读占80%，写占20%。

我们修改为读占20%，写占80%。

可以看到读的成功率进一步提高，然后写的成功率仍然保持在100%。







在这里我们发现当实现多并发的时候，机器往往会出现FAIL的情况。

# 六、实验总结

本次实验熟悉了对象存储技术，代表性系统及其特性； 实践了对象存储系统，部署了实验环境，并进行了初步测试；同时基于对象存储系统，架设了实际应用，并且完成了主要功能。虽说对Linux操作系统并不陌生，但是没有详细的了解过Linux系统，知道的也只是皮毛，一些简单应用操作罢了。在实验一的基础环境的搭建中更是摸不着头脑，不知道怎么回事的就遇到了一些问题，然后又糊里糊涂的解决这些问题。后来，问题也是层出不穷，比如镜像的捕获等，还有一些指令也不明白它的语法和语义。不过，随着实验的深入，在老师和同学的耐心指导以及自己查阅资料的帮助下，对整个平台更加熟悉，最后也完成了实验。

虽然说这次实验只完成了最简单的那条“Build-Test-Apply”线路，不过收获也还是不小，实验完成的成果在以后的生活中还能自行使用，希望在以后的学习中能有更大的收获。

# 参考文献

[1] ARNOLD J. OpenStack Swift[M]. O’Reilly Media, 2014.

[2] ZHENG Q, CHEN H, WANG Y等. COSBench: A Benchmark Tool for Cloud Object Storage Services[C]//2012 IEEE Fifth International Conference on Cloud Computing. 2012: 998–999.

[3] WEIL S A, BRANDT S A, MILLER E L等. Ceph: A Scalable, High-performance Distributed File System[C]//Proceedings of the 7th Symposium on Operating Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2006: 307–320.